# 第 6 章

# 隠れマルコフモデルの理論と実現法

## 6.1 序論

4 章、5 章では、音声認識に対するパターン認識アプローチの 1 つであるテンプレート法を紹介した。テンプレート法における 1 つのキーアイデアは、各パターン (例えば単語) に対する典型的な音声フレーム系列を、いくつかの平均化処理を使って導き、パターンを比較するために局所的なスペクトル距離尺度を用いることである。もう 1 つのキーアイデアは、同一話者がある単語を繰り返し発声したときや、異なる話者が同じ単語を発声したときの速度の違いに対処するために、時間的にパターンを対応づける動的計画法を用いることである。テンプレートアプローチの方法はかなり改良されており、様々な実用的応用に対し、良い認識性能を示している。

しかし、テンプレートアプローチは、統計的信号モデルの考え方に厳密な意味で基づいていない。参照パターンを生成する際のクラスタリングに統計的手法が広く用いられているが、テンプレートアプローチは、複数の参照トークン (または系列) を用いて異なる発声ごとの違いを表現しており、簡略化されたノンパラメトリックな手法として分類するのが適当である。よって、テンプレート表現に内在する統計的な信号の特徴づけは間接的に過ぎず、しばしば不適切でもある。例えば、次数を打ち切ったケプストラム歪み尺度を、テンプレートマッチングの局所的な距離尺度として使うことを考えてみよう。ケプストラム距離尺度のユークリッド距離形式は、参照ベクトルが、仮定されるいくつかの分布の平均値であるとみなすことができる。この十分統計量[1]の簡略形 (平均参照ベクトルのみを使うこ

[1] 十分統計量とは、ある確率過程のパラメータを推定するために必要なすべての情報を含む測定値の集合である。





と) は、明らかに、2 次統計量、つまり共分散を無視している。この共分散はあとでわかるように、統計モデルにおいて特に重要である (この分布は、フレームレベルにおけるケプストラム係数の変動を考慮するために使う。これにより、比較するパターンの適切なフレームどうしが適合するように時間正規化を行なうことができる)。明らかに、より精密で解析的な統計手法を音声認識に対して使う必要がある。

この章では、良く知られ、広く使われている手法で、各フレームのパターンのスペクトル特性を表現するための統計的手法、すなわち隠れマルコフモデル(Hidden Markov Model: HMM) アプローチについて勉強する (このモデルは情報通信の本では、マルコフ情報源またはマルコフ連鎖の確率関数とも言われている)。HMM の根本的な仮定として (または、他のいかなるタイプの統計モデルにおいても)、音声信号がパラメトリックな不規則過程として十分に表現でき、確率過程のパラメータが、正確できちんと定義された手法で決定 (推定) できるということがある。我々は、HMM 法が、広範囲の応用に対し、自然で信頼度の高い音声認識手法をもたらし、タスクの構文論や意味論と協調しながらシステムにうまく統合できることを示す。

隠れマルコフモデルの基本的な理論は、1960 年代後半および 1970 年代初期に Baumらが発表した一連の古典的論文 [1]-[5] に見ることができる。そして、1970 年代に CMU の Baker [6]、IBM の Jelinekら [7]-[13] によって音声処理応用において実現された。

この章では、マルコフ連鎖の理論を復習することから始め、いくつかの簡単な例を使って HMM の考え方に拡張する。IDA (Institute for Defence Analyses) の Jack Fergusonによって講義や文献 [14] で紹介された、今や古典的となったアプローチに基づき、HMM 設計のための 3 つの基本問題に焦点を置く。3 つの問題とは、1) ある特定の HMM が与えられたときに、観測系列の確率 (あるいは尤度) を評価する方法、2) 最適なモデル状態系列を決定する方法、3) 観測信号を最適に表現できるようにモデルパラメータを調整する方法、である。これらの 3 つの基本問題を解決できれば、HMM を音声認識のいくつかの問題に直ちに適用できることを示す。

## 6.2 離散時刻マルコフ過程

図 6.1 にあるように、$\{1,2,\ldots,N\}$ のように番号づけられた $N$ 個の異なる状態集合のうちのいずれかに、常に存在するような系を考える (この図では簡単のため $N=5$ とする)。 系は、等間隔の離散時刻ごとに、各状態に付与した確率に従って状態間を遷移する (同じ状態に戻ってくる場合もある)。 状態が変化する時点を $t = 1,2,\ldots,$ と表し、時刻 $t$ における実際の状態を $q_t$ と表記する。 一般に、上記の系を確率的に完全に記述するためには、現在 (時刻 $t$) の状態と共に過去のすべての状態を記述する必要がある。 離散時刻 1 次マルコフ連鎖のような特殊な場合は、確率的依存関係を直前の状態のみに依存するように打ち切る。つまり、

$$P[q_t = j | q_{t-1} = i,\ q_{t-2} = k, \ldots]$$

$$= P[q_t = j | q_{t-1} = i] \tag{6.1}$$

さらに、式 (6.1) の右辺が時間に独立な過程だけを考える。このとき、状態遷移確率 $a_{ij}$ の集合は次のようになる。

$$a_{ij} \ = \ P[q_t = j | q_{t-1} = i], \ \ 1 \le i,j \le N \tag{6.2}$$

これらは通常の確率的制約に従うので次の条件を満たす。

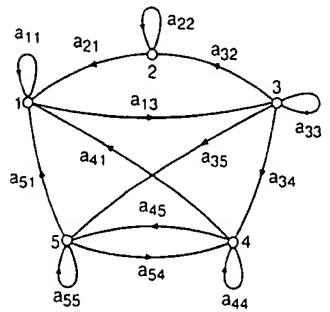



**図 6.1** 部分的な状態間で遷移のある 5 状態 (1 から 5 にラベルづけされている) のマルコフ連鎖



$$a_{ij} \ge 0 \quad \forall j,i \tag{6.3a}$$

$$\sum_{j=1}^{N} a_{ij} \ = \ 1 \quad \forall i \tag{6.3b}$$

上記の確率過程の出力は、各時刻に対応した状態の系列であり、各状態が可観測な事象と対応しているので、この過程は可観測なマルコフモデルと呼ぶことができる。 考え方を明確にするために、図 6.2 に示す簡単な 3 状態の天気のマルコフモデルを考える。 1 日に 1 回だけ (例えば正午に)、以下に示す状態の 1 つとして天気が観測されるとする。

**状態1**： 降水(雨か雪)
**状態2**： 曇り
**状態3**： 晴れ

日付 $t$ の天気は上記 3 つの状態の内の 1 つに決定されるとし、状態遷移確率行列 $A$ が次のようであるとする。

$$A \ = \ \{a_{ij}\} \ = \ \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$$

図 6.2 のモデルが与えられると、天気の時間的なパターンに関して、いくつか興味ある問題を出す (そして答える) ことができる。 例えば、次のような簡単な問題を出題できる。

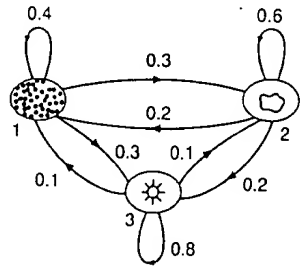



**図 6.2** 天気のマルコフモデル

**問題**

(このモデルで) 連続した 8 日間の天気が、"晴れ - 晴れ - 晴れ - 雨 - 雨 - 晴れ - 曇り - 晴れ" の確率はいくらか。

**解答**

8 日間に渡る天気の仮の集合として観測系列 O を次のように定義する。我々は図 6.2 のモデルが与えられたときの観測系列 O の確率 $P(\mathrm{O}|\text{モデル})$ を計算したい。

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| O | = | ( 晴れ, | 晴れ, | 晴れ, | 雨, | 雨, | 晴れ, | 曇り, | 晴れ ) |
| | = | ( 3, | 3, | 3, | 1, | 1, | 3, | 2, | 3 ) |
| 日付 | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

$P(\mathrm{O}|\text{モデル})$ は以下のように直接、計算できる。

$$\begin{aligned} P(\mathrm{O}|\text{モデル}) &= P[3,3,3,1,1,3,2,3|\text{モデル}] \\ &= P[3]\,P[3|3]^2 P[1|3]\,P[1|1] \\ &\quad P[3|1]\,P[2|3]\,P[3|2] \\ &= \pi_3 \cdot (a_{33})^2\, a_{31}\, a_{11}\, a_{13}\, a_{32}\, a_{23} \\ &= (1.0)(0.8)^2(0.1)(0.4)(0.3)(0.1)(0.2) \\ &= 1.536 \times 10^{-4} \end{aligned}$$

ただし、初期状態確率を

$$\pi_i = P[q_1 = i], \quad 1 \le i \le N \tag{6.4}$$

と表す。

我々が出題できる (そしてモデルを使って答えられる) 別の興味ある問題には、次のようなものもある。

**問題**

モデルの状態が既知であるとき、その状態がちょうど $d$ 日だけ続く確率はいくらか。



**解答**

この確率は、モデルが与えられたとき、次の観測系列 O の確率として計算できる。

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| O | = ( $i$, | $i$, | $i$ | ,..., | $i$, | $j \neq i$ ) | |
| 日付 | | 1 | 2 | 3 | $d$ | $d+1$ | |

$$\begin{aligned} &P(\mathrm{O}|\text{モデル},\ q_1 = i) \\ &\quad = P(\mathrm{O}, q_1 = i|\text{モデル})/P(q_1 = i) \\ &\quad = \pi_i (a_{ii})^{d-1}(1 - a_{ii})/\pi_i \\ &\quad = (a_{ii})^{d-1}(1 - a_{ii}) \\ &\quad = p_i(d) \end{aligned} \tag{6.5}$$

$p_i(d)$ は状態 $i$ における継続時間 $d$ の確率分布関数である。マルコフ連鎖の状態継続長の特性はこの指数分布になる。$p_i(d)$ に基づいて、各状態の観測事象数 (継続時間長) の期待値を、始点がその状態中であることを条件に次の様に計算できる。

$$\bar{d}_i = \sum_{d=1}^{\infty} d\, p_i(d) \tag{6.6a}$$

$$= \sum_{d=1}^{\infty} d\,(a_{ii})^{d-1}\,(1 - a_{ii}) = \frac{1}{1 - a_{ii}} \tag{6.6b}$$

よって、モデルに従えば、晴れが連続する日数の期待値は $1/(0.2) = 5$ 日、曇りは 2.5 日、雨は 1.67 日である。

**問題**

$p_i(d)$ の平均値の式、つまり式 (6.6b) を導出せよ。

**解答**

$$\bar{d}_i = \sum_{d=1}^{\infty} d\, p_i(d)$$

$$
\begin{aligned}
&= \sum_{d=1}^{\infty} d\,(a_{ii})^{d-1}(1-a_{ii}) \\
&= (1-a_{ii})\,\frac{\partial}{\partial a_{ii}}\left[\sum_{d=1}^{\infty} a_{ii}^{d}\right] \\
&= (1-a_{ii})\,\frac{\partial}{\partial a_{ii}}\left(\frac{a_{ii}}{1-a_{ii}}\right) \\
&= \frac{1}{1-a_{ii}}
\end{aligned}
$$

## 6.3 隠れマルコフモデルへの拡張

これまで我々は、各状態が決定的に観測可能な観測事象に対応するマルコフモデルを考えてきた。 よって、そのような情報源はどの状態にあるときでも、出力は不規則ではない。 このモデルは、興味ある多くの問題に対し適用するには制限が厳しすぎる。 この節では、マルコフモデルの概念を、観測事象が状態に依存した確率関数である場合も含むように拡張する。つまり、結果として得られるモデル(隠れマルコフモデル(Hidden Markov Model: HMM)と呼ばれる)は2重の確率過程になっており、背後にある確率過程は、直接には観測できず(隠れている)、観測系列を生成するもう1つの確率過程の集合を通してのみ観測できる。

コイン投げ実験を含むいくつかの簡単な例を使って、隠れマルコフモデルの基本的な概念をわかりやすく説明する。 まず、確率について、いくつかの基本的な概念を復習するために次の練習問題から始める。

**練習 6.1**

1枚の公正なコイン、つまり $P(\text{表}) = P(\text{裏}) = 0.5$ であるコインを1回投げ、表裏を観測する。

a) 10回投げて(表表裏表裏裏表裏裏表)の系列になる確率はいくらか。

b) 10回投げて(表表表表表表表表表表)の系列になる確率はいくらか。

c) 10回のうち5回が裏になる確率はいくらか。 また、10回投げるうち、裏が出ると期待される回数はいくらか。



**解答 6.1**

a) コインの表裏の出る確率が公正で、コイン投げが毎回独立して行なわれるなら、10回投げたときのいかなる観測系列の確率も $(1/2)^{10}$ になる。なぜなら、系列の数は全部で $2^{10}$ 個あり、それらが等しく起こりうるからである。つまり、

$$P(\text{表表裏表裏裏表裏裏表}) = \left(\frac{1}{2}\right)^{10}$$

b)

$$P(\text{表表表表表表表表表表}) = \left(\frac{1}{2}\right)^{10}$$

よって、表または裏だけが連続する長さ10の任意の系列は、表と裏が組み合わさった系列と同じ確率で起こる。

c) 10回のうち5回が裏になる確率はちょうど、5回が裏で5回が表(順序は問わない)になる観測系列の数である。これは、

$$P(\text{表が5回, 裏が5回}) = \binom{10}{5}\left(\frac{1}{2}\right)^{10} = \frac{252}{1024} \cong 0.25$$

なぜなら、10回投げて表が5回、裏が5回となる場合は $\binom{10}{5}$ 通りあり、それぞれの系列の確率は $\left(\frac{1}{2}\right)^{10}$ であるからである。 10回投げて裏が出ると期待される回数は、

$$E(\text{10回投げて裏が出る回数}) = \sum_{d=0}^{10} d\binom{10}{d}\left(\frac{1}{2}\right)^{10} = 5$$

よって、コインを10回投げると、平均して表が5回、裏が5回出ると期待できるが、実際にちょうど表が5回、裏が5回になる確率はたった0.25である。

### 6.3.1 コイン投げモデル

次のような場合を考える。 いまあなたは、何が起こっているのかわからないように(例えばカーテンなどで)仕切られた部屋の中にいる。 仕切りの反対側では別の人が1個(または複数個)のコインを使ってコイン投げの実験をしている。 その人はあなたに各コイン投げの結果(表か裏か)だ

けを伝え、毎回の試行でどのコインを選んだかは伝えない。このようにして、表と裏が連続した観測系列を生成するコイン投げ実験が隠れて行なわれている。観測系列は例えば、次のようであるだろう。

$$\begin{aligned} O &= (o_1 o_2 o_3 \ldots o_T) \\ &= (\text{表表裏裏裏表裏裏表} \ldots \text{表}) \end{aligned}$$

問題は、この様な状況において、表と裏の観測系列を説明する(モデル化する)HMMをどの様に作成するかである。まず、モデルの状態を何に対応させるか、そして、状態数をいくつにするかという問題に直面する。1つの可能な選択としては、たった1枚の偏ったコイン[2]が使われていると仮定することである。この場合、各状態がコイン投げの出力(つまり、表か裏)に対応する2状態モデルで状況をモデル化することができる。このモデルは図6.3aのようになる。この場合、マルコフモデルは観測可能(observable)で、モデルを完成するためには、ただ1つのモデルパラメータ(例えば、表の出る確率)の最適値を決定すればよい。面白いことに、図6.3aと同等のHMMを、縮退した1状態モデルで表現することもできる。この場合、状態は1枚の偏ったコインに対応し、未知パラメータはコインの偏り度(bias)である。

コイン投げの結果の観測系列を説明する2番目のHMMは、図6.3bで与えられる。この場合、モデルは2状態で、それぞれの状態は、異なる偏ったコインに対応する。それぞれの状態は、表と裏の出る確率分布によって特徴づけられ、状態間の遷移は状態遷移行列で特徴づけられる。どのように状態遷移が選ばれるかの物理的なメカニズムは、独立して繰り返し行なわれるコイン投げなどの確率事象に従う。

コイン投げの結果の観測系列を説明する3番目のHMMは、図6.3cで与えられる。このモデルは、3枚の偏ったコインを使い、ある確率事象に基づいて、3枚のうちから1枚を選ぶことに対応する。

図6.3に示す表と裏の観測系列を説明する3つのモデルから1つを選択する場合、問題はどのモデルが一番、実際の観測に適合するかということであろう。図6.3aの簡単な1枚のコインモデルにはただ1つの未知パラメータが、図6.3bの2枚のコインモデルには4つの未知パラメータが、そ

[2]訳注: 偏ったコインとは表と裏の出る確率が等しくないコインのことを指す。



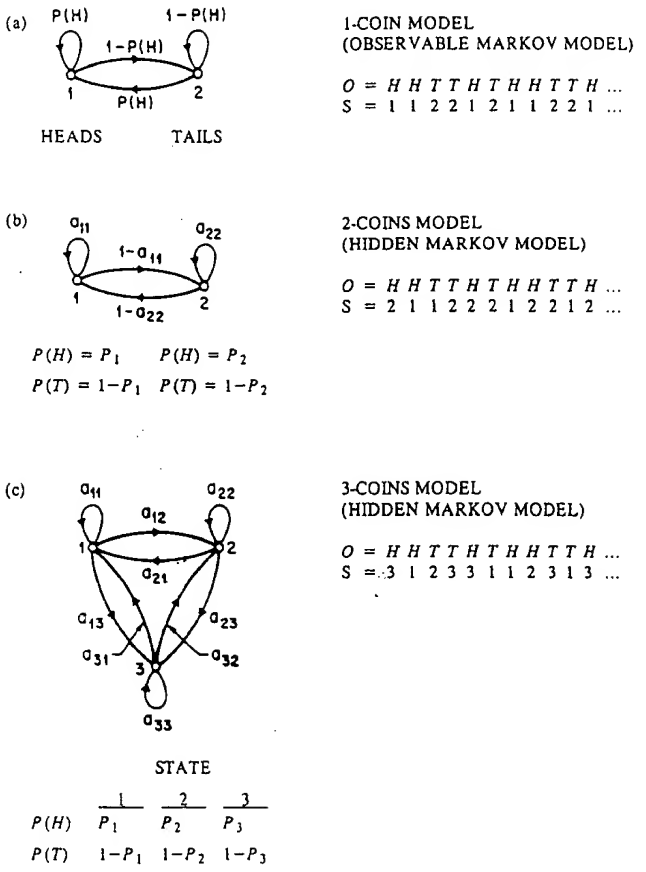



図 6.3 隠れて行なわれるコイン投げ実験の結果を説明できる3つのマルコフモデル。(a)1枚コインモデル(1-coin model)、(b) 2枚コインモデル(2-coins model)、(c) 3枚コインモデル(3-coins model)

して、図6.3cの3枚コインモデルには9つの未知パラメータが存在することは明らかである。大きなHMMは、より自由度が大きいので、それと同等のより小さなHMMよりもコイン投げ実験の系列をモデル化する能力が大きいように思えるかもしれない。これは理論的には正しいが、この章のあとの方でわかるように、実際には、我々が考えることのできるモデルの大きさにはいくつかの強い制約がある。ここでの基礎的な問題は、表-裏の観測系列が、複雑なモデルを決定できるほど十分に長く存在するかど

うかである。またこれは、ひょっとしたら、たった1枚のコインを投げた場合かもしれない。このとき、図6.3cの3枚コインモデルを使うのは、十分に能力を発揮することができない系を使うことになるので不適切であろう。

### 6.3.2 つぼとボールモデル

HMMの考え方をより複雑な状況に拡張するために、図6.4のつぼ(urn)とボールの系を考える。いま、$N$個(大きな数)の硝子のつぼが部屋の中にあるとする。それらの中には$M$種類の色のついた、たくさんのボールが入っているとする。観測事象を得るための物理的な過程は以下のようである。部屋の中に妖精がいて、ある不規則な手段に従って初めのつぼを選ぶ。そのつぼから1個のボールを無作為に選び、そのボールの色を観測事象として記録する。そのボールは、選んだつぼの中に戻される。新しいつぼが、いま選ばれたつぼに関連した不規則な選択手段に従って選ばれる。この様にして、ボール選択過程が繰り返される。この過程全体によって、有限個のボールの色の観測系列が生成される。我々はこれを、HMMの可観測な出力としてモデル化したい。

つぼとボール過程に対応する最も簡単なHMMは、各状態がそれぞれのつぼに対応し、それぞれの状態ごとにボールの色に対する確率が定義されているHMMであることは明らかだろう。つぼの選択はHMMの状態遷移行列で記述される。

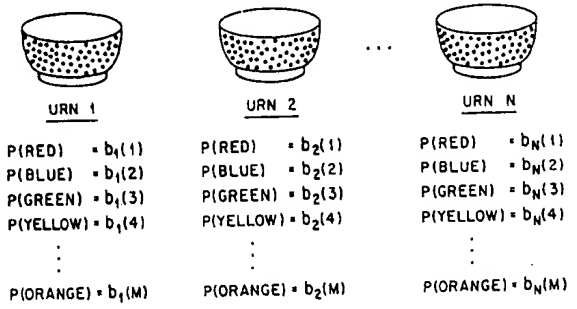



**図 6.4** 離散シンボルHMMの一般的な場合を説明する$N$状態のつぼとボールモデル



それぞれのつぼの中にあるボールの色の種類は同じであるかもしれないが、色のついたボールの数の構成が違う可能性があることに注意してほしい。よって、個別に観測して得られたボールの色から、それが取り出されたつぼを即座に特定することはできない。

### 6.3.3 HMMの要素

上記の例によって我々は、HMMが何であるか、HMMがどの様にいくつかの簡単な場面に適用できるか、その概念を得ることができる。ここで、HMMの要素を正式に定義する。

上記のつぼとボールモデルのような離散シンボル観測事象のHMMは、次の様に表現される。

1) $N$：モデル中の状態数。実際の応用では多くの場合、状態は隠れているが、モデルの状態、もしくは状態の集合に、ある物理的な意味が付けられることが多い。コイン投げ実験では、各状態は個々の偏ったコインに対応する。つぼとボールモデルでは、状態はつぼに対応する。一般に、すべての状態は他のあらゆる状態に到達できるように連結されている(すなわち、エルゴディック(ergodic)モデル)。しかしながら、この章のあとの方でわかるように、他の形態の状態連結の方が重要で、音声応用ではより適切であることが多い。各状態に$\{1,2,\ldots,N\}$とラベルを付与し、時刻$t$における状態を$q_t$と表す。

2) $M$：各状態における異なる観測シンボルの数。つまり、離散アルファベット(シンボルの種類)の大きさ。観測シンボルはモデル化する系の物理的な出力に対応する。コイン投げ実験では単に表と裏であり、ボールとつぼのモデルでは、つぼから選ばれるボールの色である。個々のシンボルを$V=\{\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_M\}$と表す。

3) 状態遷移確率行列$A=\{a_{ij}\}$、ここで、

$$a_{ij} = P[q_{t+1}=j|q_t=i], \quad 1\le i,j\le N \tag{6.7}$$

あらゆる状態が1ステップであらゆる他の状態に到達できるような特別な場合では、すべての$i,j$に対して$a_{ij}>0$となる。他の形態のHMMでは、1つまたは複数個の$(i,j)$の組に対し$a_{ij}=0$となることもある。

4) 観測シンボル確率分布$B=\{b_j(k)\}$、ここで、

$$b_j(k) = P[\mathbf{o}_t=\mathbf{v}_k|q_t=j], \quad 1\le k\le M \tag{6.8}$$

これは状態 $j$, $(j = 1, 2, \ldots, N)$ におけるシンボル分布を定義する。

5) 初期状態分布 $\pi = \{\pi_i\}$、ここで、

$$\pi_i = P[q_1 = i], \quad 1 \leq i \leq N \tag{6.9}$$

上記の議論から、HMM を完全に記述するためには、$N$、$M$ の 2 つのモデルパラメータ、観測シンボル、そして $A$、$B$、$\pi$ の 3 つの確率尺度の集合を決定する必要があることがわかる。簡単のため、モデルのパラメータ集合全体を示すために、次の簡潔な表記を用いる。

$$\lambda = (A, B, \pi) \tag{6.10}$$

もちろん、このパラメータ集合を使って O に対する確率尺度、つまり $P(\mathbf{O}|\lambda)$ を定義することができる。これについては次の節で議論する。我々は HMM という用語を、パラメータ集合 $\lambda$ と、関連する確率尺度の両方を指し示すために用いるが、紛らわしいことはないだろう。

### 6.3.4 観測事象の HMM 生成器

$N$、$M$、$A$、$B$、$\pi$ に適当な値が与えられたとき、HMM は観測系列

$$\mathbf{O} = (\mathbf{o}_1 \mathbf{o}_2 \ldots \mathbf{o}_T) \tag{6.11}$$

の生成器として以下のように使うことができる (ここで、各観測事象 $\mathbf{o}_t$ は $V$ の中のシンボルの 1 つである。そして、$T$ は観測事象の系列中の事象の数である)。

1) 初期状態分布 $\pi$ に従い初期状態 $q_1 = i$ を選ぶ。

2) $t = 1$ を代入する。

3) 状態 $i$ のシンボル確率分布 $b_i(k)$ に従い、$\mathbf{o}_t = \mathbf{v}_k$ を選ぶ。

4) 状態 $i$ の状態遷移確率分布 $a_{ij}$ に従い、新しい状態 $q_{t+1} = j$ に遷移する。

5) $t = t + 1$ を代入する。$t < T$ ならばステップ 3) に戻る。そうでなければ手続きを終了する。

次の表は、上記の手続きによって生成された状態と観測事象の系列を示す。



| 時刻, $t$ | 1 | 2 | 3 | 4 | 5 | 6 | ... | $T$ |
|---|---|---|---|---|---|---|---|---|
| 状態 | $q_1$ | $q_2$ | $q_3$ | $q_4$ | $q_5$ | $q_6$ | ... | $q_T$ |
| 観測事象 | $\mathbf{o}_1$ | $\mathbf{o}_2$ | $\mathbf{o}_3$ | $\mathbf{o}_4$ | $\mathbf{o}_5$ | $\mathbf{o}_6$ | ... | $\mathbf{o}_T$ |

上記の手続きは、観測事象の生成器として、あるいは与えられた観測系列が、適切な HMM によってどの様に生成されたかをシミュレートするモデルとしても使うことができる。

**練習 6.2**

コイン投げ実験の HMM 表現 ($\lambda$ でパラメータ化する) を考える。次の確率を持つ 3 状態モデル (3 つの異なるコインに対応する) を仮定する。

| | 状態 1 | 状態 2 | 状態 3 |
|---|---|---|---|
| $P$(表) | 0.5 | 0.75 | 0.25 |
| $P$(裏) | 0.5 | 0.25 | 0.75 |

すべての状態遷移確率は等しく 1/3 であるとする (初期状態確率は 1/3 を仮定する)。

1) つぎの系列を観測するとき、

$$\mathbf{O} = (\text{表 表 表 表 裏 表 裏 裏 裏 裏})$$

一番もっともらしい状態系列を求めよ。上記の観測系列と、一番もっともらしい状態系列の同時確率を求めよ。

2) この観測系列がすべて状態 1 から出力される確率を求めよ。

3) 次の観測系列を考えるとき、1) と 2) の答はどう変わるか。

$$\tilde{\mathbf{O}} = (\text{表 裏 裏 表 裏 表 表 裏 裏 表})$$

4) 以下に示す状態遷移確率を持つ新しいモデル $\lambda'$ の場合、1) から 3) の答はどう変わるか。これは、モデルから生成される系列のタイプについて何を示唆するか。

$$a_{11} = 0.9 \quad , \quad a_{21} = 0.45 \quad , \quad a_{31} = 0.45$$
$$a_{12} = 0.05 \quad , \quad a_{22} = 0.1 \quad , \quad a_{32} = 0.45$$
$$a_{13} = 0.05 \quad , \quad a_{23} = 0.45 \quad , \quad a_{33} = 0.1$$

**解答 6.2**

1) $\mathbf{O}$ = (表表表表裏表裏裏裏裏) が与えられ、すべての状態遷移が等しく起こる場合、一番もっともらしい状態系列は、個々の観測事象に対する確率が最大となる系列である。 表に対して一番もっともらしい状態は 2 で、裏に対して一番もっともらしい状態は 3 である。 よって、一番もっともらしい状態系列は、

$$\mathbf{q} = (2222323333)$$

である。(モデルが与えられたときの) $\mathbf{O}$ と $\mathbf{q}$ の同時確率は、

$$P(\mathbf{O}, \mathbf{q}|\lambda) = (0.75)^{10}\left(\frac{1}{3}\right)^{10}$$

2) $\hat{\mathbf{q}}$ が次のように与えられたときの $\mathbf{O}$ の確率は、

$$\hat{\mathbf{q}} = (1111111111)$$

$$P(\mathbf{O}, \hat{\mathbf{q}}|\lambda) = (0.50)^{10}\left(\frac{1}{3}\right)^{10}$$

$P(\mathbf{O}, \mathbf{q}|\lambda)$ の $P(\mathbf{O}, \hat{\mathbf{q}}|\lambda)$ に対する比は、

$$R = \frac{P(\mathbf{O}, \mathbf{q}|\lambda)}{P(\mathbf{O}, \hat{\mathbf{q}}|\lambda)} = \left(\frac{3}{2}\right)^{10} = 57.67$$

予想されたようにこの結果は、$\mathbf{q}$ が $\hat{\mathbf{q}}$ よりももっともらしいことを示している。

3) 表と裏の数が等しい $\tilde{\mathbf{O}}$ が与えられた場合でも、1) と 2) の答は変わらない。なぜなら、一番もっともらしい状態は、両者とも同じ回数だけ起こるからである。

4) $\mathbf{O}$ と $\mathbf{q}$ の新しい同時確率は、

$$P(\mathbf{O}, \mathbf{q}|\lambda') = (0.75)^{10}\left(\frac{1}{3}\right)(0.1)^6(0.45)^3$$

$\mathbf{O}$ と $\hat{\mathbf{q}}$ の新しい同時確率は、

$$P(\mathbf{O}, \hat{\mathbf{q}}|\lambda') = (0.50)^{10}\left(\frac{1}{3}\right)(0.9)^9$$

それらの比は、



$$R = \left(\frac{3}{2}\right)^{10}\left(\frac{1}{9}\right)^6\left(\frac{1}{2}\right)^3 = 1.36 \times 10^{-5}$$

言い換えると、状態遷移確率が一様ではないので、$\hat{\mathbf{q}}$ が $\mathbf{q}$ よりももっともらしい (より理解を深めるために、読者は一番もっともらしい状態系列を求めよ)。 もはや、$\tilde{\mathbf{O}}$ と $\mathbf{q}$ の確率は $\mathbf{O}$ と $\mathbf{q}$ の確率と等しくない。つまり、

$$P(\tilde{\mathbf{O}}, \mathbf{q}|\lambda') = \frac{1}{3}(0.1)^6(0.45)^3(0.25)^4(0.75)^6$$

$$P(\tilde{\mathbf{O}}, \hat{\mathbf{q}}|\lambda') = (0.50)^{10}\left(\frac{1}{3}\right)(0.9)^9$$

比は、

$$R = \left(\frac{1}{9}\right)^6\left(\frac{1}{2}\right)^3\left(\frac{1}{2}\right)^4\left(\frac{3}{2}\right)^6 = 1.67 \times 10^{-7}$$

$a_{11} = 0.9$ なので、明らかに $\hat{\mathbf{q}}$ の方がもっともらしい。

## 6.4 HMM の 3 つの基本問題

前節のような HMM の形式が与えられたとき、モデルを実世界の応用に役立つようにするためには、以下に示す 3 つの重要な基本問題を解決しなければならない。

**問題 1:** 観測系列 $\mathbf{O} = (\mathbf{o}_1\mathbf{o}\ldots\mathbf{o}_T)$ とモデル $\lambda = (A, B, \pi)$ が与えられたとき、モデルに対する観測系列の確率 $P(\mathbf{O}|\lambda)$ をいかに効率良く計算するか。

**問題 2:** 観測系列 $\mathbf{O} = (\mathbf{o}_1\mathbf{o}\ldots\mathbf{o}_T)$ とモデル $\lambda$ が与えられたとき、ある意味において最適な (例えば、観測事象を最も良く "説明" する) 状態系列 $\mathbf{q} = (q_1q_2\ldots q_T)$ をいかに選ぶか。

**問題 3:** $P(\mathbf{O}|\lambda)$ を最大にするために、モデルパラメータ $\lambda = (A, B, \pi)$ をいかに調整するか。

問題 1 は評価の問題である。すなわち、モデルと観測系列が与えられたとき、観測系列がそのモデルによって生成された確率をどの様に計算するかである。 この問題は、与えられたモデルが、与えられた観測系列に対してどの程度、適合するかのスコアリング問題としても見ることができる。後者の見方は、たいへん有益である。 例えば、競合するいくつかのモデルから 1 つを選択する場合に、問題 1 の解によって、観測事象に最も良く適

合するモデルを選ぶことができる。

問題2は、モデルの隠れた部分を明らかにしようとする問題である。つまり、"正しい" 状態系列を求めることである。縮退 (degenerate) モデルの場合を除いたすべてのモデルにおいて、求めるべき "正しい" 状態系列は存在しないことは明らかであろう。よって、実際には、通常、この問題をできるだけ適切に解決するために、ある最適化基準を用いる。あとでわかるように、いくつかの適切な最適化基準が導入できるので、求める状態系列の使用目的によって、基準を選択する必要がある。典型的な使用目的は、モデルの構造を学習すること、連続音声認識の最適状態系列を見つけ出すこと、各状態の平均統計量を得ることなどであろう。

問題3は、与えられた観測系列がいかに生成されたかを最も良く記述するように、モデルパラメータを最適化するものである。モデルパラメータを調整するために使われる観測系列は、HMMを "学習" するために使われるので、学習系列と呼ばれる。ほとんどのHMMの応用において学習問題は極めて重要である。なぜなら、観測された学習データに対して、モデルパラメータを最適に適応できる、すなわち、実際の現象に対して最も良いモデルを生成できるからである。

考え方を明確にするために、次の簡単な孤立単語音声認識器を考える。いま、$W$ 個の語彙の各単語に対して $N$ 状態のHMMを設計することを考える。与えられた単語の音声信号を、符号化されたスペクトルベクトルの時系列として表現する。独立した $M$ 個のスペクトルベクトルからなるスペクトル符号帳を用いて符号化するとする。各観測事象は、もとの音声信号に (あるスペクトル歪みの意味で) 最も近いスペクトルベクトルのインデックス (引数) である。よって学習系列には、各語彙単語に対して、(1人、または複数の話者の) 符号帳インデックスの系列が数多く繰り返されたものが含まれている。最初のタスクは、各単語モデルを構築することである。これは、各単語モデルに対するモデルパラメータを最適に推定する問題3の解を用いて行なうことができる。次に、モデルの各状態の物理的な意味を理解するために、問題2に対する解を用いて、各単語学習データ系列を状態ごとに分割し、各状態で発生する観測事象のもとになるスペクトルベクトルの特性を調べる。これは、発声された単語系列をモデル化する能力が向上するように、精密なモデルを作成するために行なう (例えば、状態数を増やしたり、符号帳サイズなどを変える)。$W$ 個のHMMの集合が設計され、最適化されると、最後は未知単語の認識である。認識



は、問題1の解を用いて、与えられたテスト観測系列に対する各単語モデルのスコアを計算し、最も高いスコア (すなわち、最も高い尤度) のモデルの単語を認識結果として選択することで行なわれる。

次の節では、HMMに関する3つの基本問題に対する数学的な解の定式化を示す。3つの問題は、確率の枠組みの中でお互い密接に関連していることがわかるだろう。

### 6.4.1 問題1の解 – 確率評価 –

モデル $\lambda$ が与えられたときの、観測系列 $\mathbf{O} = (\mathbf{o}_1 \mathbf{o} \ldots \mathbf{o}_T)$ に対する確率 $P(\mathbf{O}|\lambda)$ を計算することを考える。これを実行するための最も直接的な方法は、長さ $T$ (観測系列の長さ) の可能なすべての状態系列を数え上げることである。そのような状態系列は $N^T$ 個存在する。その中の1つとして、状態系列が固定された場合を考える。

$$\mathbf{q} = (q_1 q_2 \ldots q_T) \tag{6.12}$$

ここで、$q_1$ は初期状態である。式 (6.12) で示す状態系列が与えられたとき、観測系列 $\mathbf{O}$ の確率は以下のようになる。

$$P(\mathbf{O}|\mathbf{q}, \lambda) = \prod_{t=1}^{T} P(\mathbf{o}_t|q_t, \lambda) \tag{6.13a}$$

ここでは、観測事象が統計的に独立であることを仮定している。よって、

$$P(\mathbf{O}|\mathbf{q}, \lambda) = b_{q_1}(\mathbf{o}_1) \cdot b_{q_2}(\mathbf{o}_2) \ldots b_{q_T}(\mathbf{o}_T) \tag{6.13b}$$

を得る。このような状態系列 $\mathbf{q}$ の確率は次のように書ける。

$$P(\mathbf{q}|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \cdots a_{q_{T-1} q_T} \tag{6.14}$$

$\mathbf{O}$ と $\mathbf{q}$ の同時確率、つまり、$\mathbf{O}$ と $\mathbf{q}$ が同時に起こる確率は単に上記2つの項の積である。つまり、

$$P(\mathbf{O}, \mathbf{q}|\lambda) = P(\mathbf{O}|\mathbf{q}, \lambda) P(\mathbf{q}|\lambda) \tag{6.15}$$

(モデルが与えられたときの) $\mathbf{O}$ の確率は、起こり得るすべての状態系列 $\mathbf{q}$ について、この同時確率を総和して得られる。すなわち、以下のようになる。

$$P(\mathrm{O}|\lambda) = \sum_{\text{all } \mathbf{q}} P(\mathrm{O}|\mathrm{q},\lambda)P(\mathrm{q}|\lambda) \quad (6.16)$$

$$= \sum_{q_1,q_2,\cdots,q_T} \pi_{q_1} b_{q_1}(\mathrm{o}_1) a_{q_1q_2} b_{q_2}(\mathrm{o}_2) \ldots a_{q_{T-1}q_T} b_{q_T}(\mathrm{o}_T) \quad (6.17)$$

この式中の計算は以下のように解釈できる。 初めに(時刻$t = 1$で)状態$q_1$に$\pi_{q_1}$の確率で存在し、(この状態中で)シンボル$\mathrm{o}_1$を確率$b_{q_1}(\mathrm{o}_1)$で生成する。 時刻は$t$から$t+1$(時刻$= 2$)に移り、状態$q_1$から状態$q_2$へ確率$a_{q_1q_2}$で遷移する。ここで、シンボル$\mathrm{o}_2$を確率$b_{q_2}(\mathrm{o}_2)$で生成する。 このようにしてこの過程は、最後に時刻$T$で状態$q_{T-1}$から$q_T$へ確率$a_{q_{T-1}q_T}$で遷移し、シンボル$\mathrm{o}_T$を確率$b_{q_T}(\mathrm{o}_T)$で生成するまで続けられる。

直接的な定義(式(6.17))に従えば、$P(\mathrm{O}|\lambda)$の計算に$2T \cdot N^T$のオーダーの計算が必要なことが、読者にも納得できるだろう。なぜなら、各時刻$t = 1, 2, \ldots, T$では到達可能な状態が$N$個あり(つまり、$N^T$個の可能な状態系列が存在し)、その各状態系列に対して、式(6.17)では、総和をとるべき各項に対して約$2T$個の計算が必要だからである(正確には、$(2T-1)N^T$個の積と$N^T - 1$個の和が必要である)。 この計算は、たとえ$N$や$T$が小さな値であっても計算不可能である。例えば、$N = 5$(状態)、$T = 100$(観測数)のとき、$2 \cdot 100 \cdot 5^{100} \approx 10^{72}$のオーダーの計算になってしまう! 明らかに、問題1を解くためには、より効率的な方法が必要である。 幸いなことに、そのような方法(前向き処理と呼ばれる)が存在する。

#### 6.4.1.1 前向き処理

前向き変数$\alpha_t(i)$を以下のように定義する。

$$\alpha_t(i) = P(\mathrm{o}_1\mathrm{o}_2 \ldots \mathrm{o}_t, q_t = i|\lambda) \quad (6.18)$$

これは、モデル$\lambda$が与えられたときに、部分的な観測系列$\mathrm{o}_1\,\mathrm{o}_2 \ldots \mathrm{o}_t$を(時刻$t$までに)出力し、時刻$t$に状態$i$に存在する確率である。 $\alpha_t(i)$は以下のように帰納的に計算できる。

1) 初期化:

$$\alpha_1(i) = \pi_i b_i(\mathrm{o}_1), \quad 1 \le i \le N \quad (6.19)$$

2) 帰納:



$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)\, a_{ij}\right] b_j(\mathrm{o}_{t+1}), \qquad 1 \le t \le T-1$$
$$1 \le j \le N \quad (6.20)$$

3) 終了:

$$P(\mathrm{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \quad (6.21)$$

ステップ1)では、前向き確率を、状態$i$と初期観測事象$\mathrm{o}_1$の同時確率として初期化する。 前向き計算の中心である帰納ステップを図6.5(a)で説明する。 この図は時刻$t$で到達可能な$N$個の状態$i$, $(1 \le i \le N)$から、時刻$t+1$で状態$j$にいかに到達できるかを示している。 $\alpha_t(i)$は、$\mathrm{o}_1\,\mathrm{o}_2 \ldots \mathrm{o}_t$が観測され、かつ、時刻$t$で状態$i$に存在するという同時事象の確率なので、積$\alpha_t(i)a_{ij}$は、$\mathrm{o}_1\,\mathrm{o}_2 \ldots \mathrm{o}_t$が観測され、時刻$t$で状態$i$を経た後に時刻$t+1$で状態$j$に到達する同時事象の確率である。 この積を

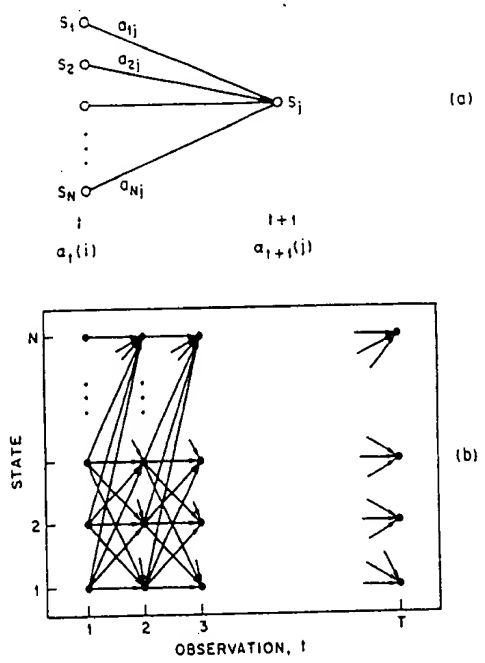



**図 6.5** (a) 前向き変数$\alpha_{t+1}(j)$の計算に必要な一連の処理の説明図、(b) 観測(observation) $t$と状態(state) $i$で作られる格子上で$\alpha_t(i)$の計算を実現する方法

時刻 $t$ における可能な $N$ 個のすべての状態 $i, (1 \le i \le N)$ で総和した結果は、時刻 $t$ 以前の部分的な観測事象のすべてを考慮して時刻 $t+1$ で状態 $j$ に存在する確率である。一旦、これが計算され、$j$ が決まれば、状態 $j$ の観測事象 $o_{t+1}$ を考慮することにより、つまり、確率 $b_j(o_{t+1})$ を、総和した結果に掛け合わせることにより、$\alpha_{t+1}(j)$ が得られることは容易に理解できる。式 (6.20) の計算は、時刻 $t$ のすべての状態 $j, (1 \le j \le N)$ に対して実行され、さらに $t = 1, 2, \ldots, T-1$ に対して繰り返される。最後に、目的である確率 $P(O|\lambda)$ は、ステップ 3) によって、前向き確率の終端の確率 $\alpha_T(i)$ を総和して求められる。なぜなら、定義から、

$$\alpha_T(i) = P(o_1 o_2 \ldots o_T, q_T = i|\lambda) \tag{6.22}$$

であるから、$P(O|\lambda)$ は単なる $\alpha_T(i)$ の総和になるのである。$\alpha_t(j), (1 \le t \le T, 1 \le j \le N)$ の導出に必要な計算量を調べると、$N^2T$ のオーダーであり (再び、正確には $N(N+1)(T-1)+N$ 個の積と $N(N-1)(T-1)$ 個の和)、直接的な計算で必要だった $2TN^T$ 個ではないことがわかる。$N = 5$、$T = 100$ に対して、直接的な計算では $10^{72}$ 回の計算が必要なのに対し、前向き処理では約 3000 回の計算量で済み、約 69 オーダー (桁) の計算規模の削減になる。

実際には、前向き確率の計算は、図 6.5(b) に示す格子 (あるいはトレリス (trellis)) 構造に基づいて実行される。どんなに観測系列が長くても、全部で $N$ 状態しかないので (格子の各タイムスロットにおけるノード数)、可能なすべての状態系列は、各時刻でこれら $N$ 個のノードに再び集約されることがキーポイントである。時刻 $t = 1$ (格子の最初のタイムスロット) では、$\alpha_1(i), (1 \le i \le N)$ の値を計算する必要がある。時刻 $t = 2, 3, \ldots, T$ では、$\alpha_t(j), (1 \le j \le N)$ の値だけを計算すればよい。$N$ 個の各格子点は、直前のタイムスロットの N 個の格子点からしか到達できないので、それぞれの計算は直前のたった $N$ 個の値 $\alpha_{t-1}(i)$ にのみ関係する。

#### 6.4.1.2 後ろ向き処理

同様にして、以下に定義する後ろ向き変数 $\beta_t(i)$ を考えることができる。

$$\beta_t(i) = P(o_{t+1} o_{t+2} \ldots o_T|q_t = i, \lambda) \tag{6.23}$$

これは、モデル $\lambda$ と時刻 $t$ における状態 $i$ が与えられたとき、時刻 $t+1$ から



ら終端までの部分的な観測系列の確率である。$\beta_t(i)$ も以下のように帰納的に計算できる。

1) 初期化:

$$\beta_T(i) = 1, \quad 1 \le i \le N \tag{6.24}$$

2) 帰納:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1})\, \beta_{t+1}(j),$$

$$t = T-1, T-2, \ldots, 1, \quad 1 \le i \le N \tag{6.25}$$

初期化ステップ 1) では、$\beta_T(i)$ がすべての $i$ に対して 1 になるように任意に定義する。図 6.6 にも示すステップ 2) は、以下の事柄を示す。時刻 $t$ に状態 $i$ に存在し、時刻 $t+1$ 以降の観測系列を考慮するためには、時刻 $t+1$ において到達可能なすべての状態 $j$ と、$i$ から $j$ への遷移 ($a_{ij}$ 項)、状態 $j$ の観測事象 $o_{t+1}(b_j(o_{t+1})$ 項)、状態 $j$ 以降の部分観測系列 ($\beta_{t+1}(j)$ 項) を考慮しなければならない。後ろ向き計算が、前向き計算と同様、HMM の基本問題 2 および 3 を解決するために、いかに貢献しているかあとでわかるだろう。

再び、$\beta_t(i), (1 \le t \le T, 1 \le i \le N)$ の計算には $N^2T$ のオーダーの計算

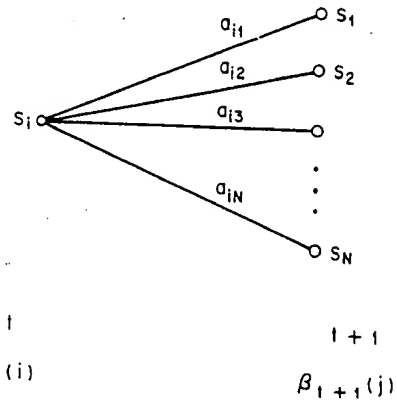



図 6.6 後ろ向き変数 $\beta_t(i)$ の計算に必要な一連の処理の説明図

が必要であるが、これは図6.5(b)と同様に格子構造上で計算可能である。

### 6.4.2 問題2の解 – "最適" 状態系列 –

厳密な解が得られる問題1と違い、問題2、すなわち、与えられた観測系列に対応する"最適"状態系列を求める問題には、いくつかの解法がある。この問題の難しさは最適状態系列の定義にある。つまり、可能ないくつかの最適性基準が存在するのである。例えば、可能な最適性基準の1つは、各時刻$t$で個々に最も起こりそうな状態$q_t$を選ぶことである。この最適性基準は、個々に正しい状態数の期待値を最大にする。この解法によって問題2を解く場合、次の事後確率変数を定義することができる。

$$\gamma_t(i) = P(q_t = i|\mathbf{O}, \lambda) \tag{6.26}$$

これは、観測系列$\mathbf{O}$とモデル$\lambda$が与えられた場合に、時刻$t$で状態$i$に存在する確率である。$\gamma_t(i)$にはいくつかの表現形式がある。例えば、次のようなものがある。

$$\begin{aligned}\gamma_t(i) &= P(q_t = i \mid \mathbf{O}, \lambda) \\ &= \frac{P(\mathbf{O}, q_t = i \mid \lambda)}{P(\mathbf{O} \mid \lambda)} \\ &= \frac{P(\mathbf{O}, q_t = i \mid \lambda)}{\sum_{i=1}^{N} P(\mathbf{O}, q_t = i \mid \lambda)}\end{aligned} \tag{6.27}$$

$P(\mathbf{O}, q_t = i \mid \lambda)$は$\alpha_t(i)\beta_t(i)$と等価なので、$\gamma_t(i)$は次のように書ける。

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)} \tag{6.28}$$

この式では、$\alpha_t(i)$は、部分的な観測系列$\mathbf{o}_1\,\mathbf{o}_2 \ldots \mathbf{o}_t$を考慮しながら、時刻$t$で状態$i$に存在する状況に対応し、$\beta_t(i)$は、時刻$t$で状態$q_t = i$に存在し、残りの観測系列$\mathbf{o}_{t+1}\mathbf{o}_{t+2} \ldots \mathbf{o}_T$を考慮することに対応している。

$\gamma_t(i)$を用いて、各時刻$t$で個々に一番もっともらしい状態$q_t^*$を、次のように求めることができる。

$$q_t^* = \underset{1 \le i \le N}{\operatorname{argmin}}[\gamma_t(i)], \quad 1 \le t \le T \tag{6.29}$$

式(6.29)は、(各時刻$t$で一番もっともらしい状態を選ぶことによって)正



しい状態数の期待値を最大にするが、結果として得られる状態系列にはいくつかの問題点がある。例えば、HMMに確率ゼロの状態遷移(ある状態$i, j$に対し$a_{ij} = 0$)が存在する場合、"最適"な状態系列は実際にはありえない状態系列であるかもしれない。なぜなら、式(6.29)の解は単に一番もっともらしい状態を各時刻で決定しているだけで、状態系列の発生確率を考慮していないからである。

上記の問題に対する1つの解決法は、最適性基準を変更することである。例えば、正しい状態の2つ組み$(q_t, q_{t+1})$あるいは3つ組み$(q_t, q_{t+1}, q_{t+2})$などが期待される数を最大にする状態系列を求めることにより解決できるかもしれない。これらの基準は、ある応用には適するが、一番広く使われる基準は、1本の最適状態系列(パス)、つまり、$P(\mathbf{q}|\mathbf{O}, \lambda)$を最大にする(これは$P(\mathbf{q}, \mathbf{O}|\lambda)$を最大にするのに等しい)パスを見つけ出す基準である。動的計画法に基づいて、この1本の最適状態系列を探索する定式化された手法があり、これをビタービ(Viterbi)アルゴリズムと呼ぶ[15, 16]。

#### 6.4.2.1 ビタービアルゴリズム

与えられた観測系列$\mathbf{O} = (\mathbf{o}_1\,\mathbf{o}_2 \ldots \mathbf{o}_T)$に対する1本の最適状態系列$\mathbf{q} = (q_1\,q_2 \ldots q_T)$を見つけるために、次の量を定義する必要がある。

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P[q_1 q_2 \ldots q_{t-1},\ q_t = i,\ \mathbf{o}_1\mathbf{o}_2 \ldots \mathbf{o}_t|\lambda] \tag{6.30}$$

$\delta_t(i)$は1本のパス上の、時刻$t$でのベストスコア(最も高い確率)である。これは、最初の$t$個の観測事象を考慮しながら、時刻$t$で状態$i$で終わっている。帰納法により、式(6.30)は次のように書ける。

$$\delta_{t+1}(j) = [\max_i \delta_t(i)\, a_{ij}] \cdot b_j(\mathbf{o}_{t+1}) \tag{6.31}$$

実際に状態系列をつないでいくために、各時刻$t$、各状態$j$で式(6.31)を最大にする経路の引数を記憶しておく必要がある。我々はこれを配列$\psi_t(j)$に記憶する。最適状態系列を求める全体的な手順は、次のように書くことができる。

1) 初期化

$$\delta_1(i) = \pi_i b_i(\mathbf{o}_1), \quad 1 \le i \le N \tag{6.32a}$$

$$\psi_1(i) = 0 \tag{6.32b}$$

2) 繰り返し:

$$\delta_t(j) = \max_{1\le i\le N}[\delta_{t-1}(i)\, a_{ij}]b_j(\mathbf{o}_t), \quad 2\le t\le T,\ 1\le j\le N \tag{6.33a}$$

$$\psi_t(j) = \operatorname*{argmax}_{1\le i\le N}[\delta_{t-1}(i)\, a_{ij}], \quad 2\le t\le T,\ 1\le j\le N \tag{6.33b}$$

3) 終了:

$$P^* = \max_{1\le i\le N}[\delta_T(i)] \tag{6.34a}$$

$$q_T^* = \operatorname*{argmax}_{1\le i\le N}[\delta_T(i)] \tag{6.34b}$$

4) パス (状態系列) バックトラック:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t=T-1,\ T-2,\ldots,1 \tag{6.35}$$

ビタービアルゴリズムは、(バックトラックステップを除いて) 式 (6.19)–(6.21) の前向き計算を実行するのに似ている点に注目してほしい。 主な相違点は、式 (6.20) で加算を行なっていたところを、式 (6.33a) では前の状態の中から最大値を選ぶ点である。 格子 (トレリス) 構造によって、ビタービ処理の計算も効率良く実行できることは明らかだろう。

#### 6.4.2.2 ビタービ実行の別手段

前節のビタービアルゴリズムは、モデルパラメータを対数化することにより、積演算を用いずに実行することができる。

**0) 前処理:**

$$\tilde{\pi}_i = \log(\pi_i), \qquad 1\le i\le N$$

$$\tilde{b}_i(\mathbf{o}_t) = \log[b_i(\mathbf{o}_t)], \quad 1\le i\le N,\ 1\le t\le T$$

$$\tilde{a}_{ij} = \log(a_{ij}), \qquad 1\le i,\ j\le N$$

**1) 初期化:**

$$\tilde{\delta}_1(i) = \log(\delta_1(i)) = \tilde{\pi}_i + \tilde{b}_i(\mathbf{o}_1), \quad 1\le i\le N$$

$$\psi_1(i) = 0, \qquad 1\le i\le N$$



**2) 繰り返し:**

$$\tilde{\delta}_t(j) = \log(\delta_t(j)) = \max_{1\le i\le N}\left[\tilde{\delta}_{t-1}(i)+\tilde{a}_{ij}\right] + \tilde{b}_j(\mathbf{o}_t)$$

$$\psi_t(j) = \operatorname*{argmax}_{1\le i\le N}\left[\tilde{\delta}_{t-1}(i)+\tilde{a}_{ij}\right], \quad 2\le t\le T,\ 1\le j\le N$$

**3) 終了:**

$$\tilde{P}^* = \max_{1\le i\le N}[\tilde{\delta}_T(i)]$$

$$q_T^* = \operatorname*{argmax}_{1\le i\le N}[\tilde{\delta}_T(i)]$$

**4) バックトラック:**

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t=T-1,\ T-2,\ldots,1$$

この別手段を実行するのに必要な計算量は、オーダー $N^2T$ の加算である (これに前処理の計算量が加わる)。 前処理は 1 度実行されればよく、その結果を保存しておけるので、そのコストはほとんどのシステムにおいて無視できるほど小さい。

**練習 6.3**

練習 6.2 のコイン投げ実験で使ったモデル (つまり、3 枚の異なるコインのモデル) が、以下の確率を持つとする。

| | 状態 1 | 状態 2 | 状態 3 |
|---|---|---|---|
| $P(H)$ | 0.5 | 0.75 | 0.25 |
| $P(T)$ | 0.5 | 0.25 | 0.75 |

すべての状態遷移確率が 1/3 で等しく、すべての初期確率が 1/3 のとき、次の観測系列

$$\mathbf{O} = (\text{表表表裏表裏裏裏裏})$$

に対する一番もっともらしいパスをビタービアルゴリズムを使って求めよ。

**解答 6.3**

$a_{ij}$ 項はすべて 1/3 なので、これらの項は除外できる (初期状態確率の項

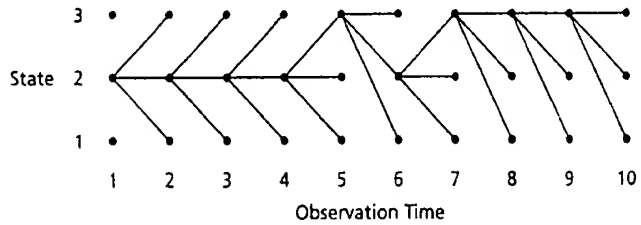
も同じく除外する)。よって、次の値を得る。

$$\delta_1(1) = 0.5,\ \delta_1(2) = 0.75,\ \delta_1(3) = 0.25$$

$\delta_t(j)$ を $(2 \le t \le 10)$ の間で繰り返すことにより次を得る。

$$\begin{array}{lll}
\delta_2(1) = (0.75)(0.5), & \delta_2(2) = (0.75)^2, & \delta_2(3) = (0.75)(0.25) \\
\delta_3(1) = (0.75)^2(0.5), & \delta_3(2) = (0.75)^3, & \delta_3(3) = (0.75)^2(0.25) \\
\delta_4(1) = (0.75)^3(0.5), & \delta_4(2) = (0.75)^4, & \delta_4(3) = (0.75)^3(0.25) \\
\delta_5(1) = (0.75)^4(0.5), & \delta_5(2) = (0.75)^4(0.25), & \delta_5(3) = (0.75)^5 \\
\delta_6(1) = (0.75)^5(0.5), & \delta_6(2) = (0.75)^6, & \delta_6(3) = (0.75)^5(0.25) \\
\delta_7(1) = (0.75)^6(0.5), & \delta_7(2) = (0.75)^6(0.25), & \delta_7(3) = (0.75)^7 \\
\delta_8(1) = (0.75)^7(0.5), & \delta_8(2) = (0.75)^7(0.25), & \delta_8(3) = (0.75)^8 \\
\delta_9(1) = (0.75)^8(0.5), & \delta_9(2) = (0.75)^8(0.25), & \delta_9(3) = (0.75)^9 \\
\delta_{10}(1) = (0.75)^9(0.5), & \delta_{10}(2) = (0.75)^9(0.25), & \delta_{10}(3) = (0.75)^{10}
\end{array}$$

これにより、次の形の(格子)図が得られる。




よって、一番もっともらしい状態系列は $\{2,2,2,2,3,2,3,3,3,3\}$ である。

### 6.4.3 問題3の解 – パラメータ推定 –

HMMの3番目の問題は、ある最適化基準を満たすようにモデルパラメータ $(A, B, \pi)$ を調整する方法を決定することである。この問題は他に比べてはるかに難しい。観測系列の確率を最大化するモデルパラメータ集合を、解析的に直接求める方法は知られていない。しかし、Baum-Welch法(EM(expectation-maxiamization)法としても知られている[17])のような繰り返し手法や、勾配法[18]によって、尤度 $P(\mathrm{O}|\lambda)$ が局所的に最大になるモデルパラメータ $\lambda = (A, B, \pi)$ を求めることはできる。この節で



は、主としてBaumらの古典的な研究に基づき、最尤(maximum likelihood: ML)モデルパラメータを選択するための繰返し手法の1つについて議論する。

てはるかに難しい。観測系列の確率を最大化するモデルパラメータ集合を、解析的に直接求める方法は知られていない。しかし、Baum-Welch法(EM(expectation-maxiamization)法としても知られている[17])のような繰り返し手法や、勾配法[18]によって、尤度 $P(\mathrm{O}|\lambda)$ が局所的に最大になるモデルパラメータ $\lambda = (A, B, \pi)$ を求めることはできる。この節で

まず、HMMパラメータの再推定手法(更新と改善の繰り返し)について述べる。初めに、モデルと観測系列が与えられたとき、時刻 $t$ に状態 $i$ に存在し、時刻 $t+1$ に状態 $j$ に存在する確率 $\xi_t(i,j)$ を定義する。すなわち、

$$\xi_t(i,j) = P(q_t = i,\ q_{t+1} = j|\mathrm{O}, \lambda) \tag{6.36}$$

式(6.36)の条件を満足するパスを図6.7に示す。前向き、後ろ向き変数の定義から、$\xi_t(i,j)$ は次のように書ける。

$$\begin{aligned}
\xi_t(i,j) &= \frac{P(q_t = i,\ q_{t+1} = j,\ \mathrm{O} \mid \lambda)}{P(\mathrm{O} \mid \lambda)} \\
&= \frac{\alpha_t(i)\ a_{ij} b_j(\mathrm{o}_{t+1})\ \beta_{t+1}(j)}{P(\mathrm{O}|\lambda)} \\
&= \frac{\alpha_t(i)\ a_{ij} b_j(\mathrm{o}_{t+1})\ \beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i)\ a_{ij} b_j(\mathrm{o}_{t+1})\ \beta_{t+1}(j)}
\end{aligned} \tag{6.37}$$

我々は以前、$\gamma_t(i)$ を、モデルと観測系列全体が与えられたときに、時刻 $t$ で状態 $i$ に存在する確率と定義した。よって、$\gamma_t(i)$ は $\xi_t(i,j)$ を $j$ について総和したものとして関係づけられる。つまり、

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \tag{6.38}$$

$\gamma_t(i)$ を時刻 $t$ について和をとれば、状態 $i$ を訪れた回数の(時間的な)期待値と見なせる値を得ることができる。それは(総和から時刻 $t = T$ を除いたならば)等価的に状態 $i$ から遷移する回数の期待値になる。同様に、$\xi_t(i,j)$ の $t$ についての($t = 1$ から $t = T-1$ までの)総和は、状態 $i$ から状態 $j$ へ遷移する回数の期待値とみなせる。つまり、

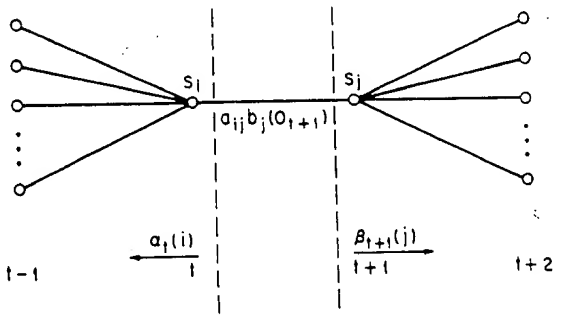



**図 6.7** 系が時刻 $t$ に状態 $i$ に存在し、時刻 $t+1$ に状態 $j$ に存在する同時事象の計算に必要な処理の流れを説明する図

$$\sum_{t=1}^{T-1}\gamma_t(i) \;=\; \mathrm{O}\text{において状態 } i \text{ から遷移する回数の期待値} \tag{6.39a}$$

$$\sum_{t=1}^{T-1}\xi_t(i,j) \;=\; \mathrm{O}\text{において状態 } i \text{ から状態 } j \text{ へ遷移する回数の期待値} \tag{6.39b}$$

上記の式 (および、事象の発生回数を数える概念) を用いて、HMM のパラメータの再推定手法を示すことができる。 $\pi, A, B$ の適切な再推定式は、次のようになる。

$$\begin{aligned}\bar{\pi}_j &= \text{時刻 } (t=1) \text{ に状態 } i \text{ に存在すると期待される頻度 (回数)} \\ &= \gamma_1(i)\end{aligned} \tag{6.40a}$$

$$\begin{aligned}\bar{a}_{ij} &= \frac{\text{状態 } i \text{ から状態 } j \text{ へ遷移する回数の期待値}}{\text{状態 } i \text{ から遷移する回数の期待値}} \\ &= \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}\end{aligned} \tag{6.40b}$$

$$\bar{b}_j(k) \;=\; \frac{\text{状態 } j \text{ にとどまりシンボル } \mathbf{v}_k \text{ を観測する回数の期待値}}{\text{状態 } j \text{ にとどまる回数の期待値}}$$



$$= \frac{\sum_{\substack{t=1 \\ \text{s.t. } O_t=\mathbf{v}_k}}^{T}\gamma_t(j)}{\sum_{t=1}^{T}\gamma_t(j)} \tag{6.40c}$$

仮に、現在のモデルを $\lambda=(A,B,\pi)$ と定義し、これを式 (6.40a)-(6.40c) の右辺を計算するために用いるとする。式 (6.40a)-(6.40c) の左辺によって決定される再推定モデルを $\bar{\lambda}=\bar{A},\bar{B},\bar{\pi}$ と定義する。Baum らは、1) 初期モデル $\lambda$ が尤度関数の臨界点(ここでは $\bar{\lambda}=\lambda$ となる点) を定義する、または、2) モデル $\bar{\lambda}$ がモデル $\lambda$ よりも $P(\mathrm{O}|\bar{\lambda}) > P(\mathrm{O}|\lambda)$ の意味で、よりもっともらしいことを証明した。つまり、観測系列が生成された可能性がより高い、新しいモデル $\bar{\lambda}$ を手に入れることができた。

上記の手法に基づいて、$\bar{\lambda}$ を $\lambda$ に入れ替えて繰り返し使いながら再推定計算を繰り返せば、O がそのモデルから観測されたという確率を、ある限界点に達するまで高めることができる。 この再推定手法の最終結果は、HMM の最尤推定値になる。 1 つ指摘しておくべきことは、前向き後ろ向きアルゴリズムは、単に極大点に導くだけであるということである。関心のあるほとんどの問題において、尤度関数はたいへん複雑で、数多くの極大点が存在する。

式 (6.40a)-(6.40c) の再推定式は、$\lambda$ に関する次の Baum の補助関数を (標準的な制約が付いた最適化手法を用いて) 最大化することにより、直接、導出することができる。

$$Q(\lambda',\lambda) \;=\; \sum_{\mathrm{q}} P(\mathrm{O},\mathrm{q}|\lambda')\log P(\mathrm{O},\mathrm{q}|\lambda) \tag{6.41}$$

なぜなら、

$$Q(\lambda',\lambda)\geq Q(\lambda',\lambda') \Rightarrow P(\mathrm{O}|\lambda)\geq P(\mathrm{O}|\lambda') \tag{6.42}$$

の関係があるので、尤度 $P(\mathrm{O}|\lambda)$ を増加させるという意味で、$\lambda$ に関する関数 $Q(\lambda',\lambda)$を $\lambda'$ を改善しながら最大化できるのである。 この手法を繰り返せば、尤度関数は最終的に臨界点に収束する。

#### 6.4.3.1 $Q$ 関数からの再推定式の導出

式 (6.41) では補助関数 $Q(\lambda',\lambda)$ を、次のように定義した。

$$Q(\lambda',\lambda) \;=\; \sum_{\mathrm{q}} P(\mathrm{O},\mathrm{q}|\lambda')\;\log P(\mathrm{O},\mathrm{q}|\lambda)$$

ここで、$P$ と $\log P$ は (HMM パラメータを用いて) 次のように表現できる。

$$P(\mathrm{O},\mathrm{q}|\lambda) = \pi_{q_0}\prod_{t=1}^{T} a_{q_{t-1}q_t} b_{q_t}(\mathrm{o}_t)$$

$$\log P(\mathrm{O},\mathrm{q}|\lambda) = \log \pi_{q_0} + \sum_{t=1}^{T} \log a_{q_{t-1}q_t} + \sum_{t=1}^{T} \log b_{q_t}(\mathrm{o}_t)$$

(上記の式と式 (6.17) の表現には多少の違いがある。式 (6.17) では、最初の観測は、状態が遷移する前の初期状態に関係していた。この違いは矛盾したものではなく、手法の理解を妨げるものではない) よって、$Q(\lambda',\lambda)$ は次のように書ける。

$$Q(\lambda',\lambda) = Q_\pi(\lambda',\pi) + \sum_{i=1}^{N} Q_{a_i}(\lambda',\mathrm{a}_i) + \sum_{i=1}^{N} Q_{b_i}(\lambda',\mathrm{b}_i)$$

ここで、$\pi = [\pi_1,\pi_2,\ldots,\pi_N]$ 、$\mathrm{a}_i = [a_{i1},a_{i2},\ldots,a_{iN}]$、$\mathrm{b}_i$ は $b_i(\cdot)$ を定義するパラメータベクトルである。そして、

$$Q_\pi(\lambda',\pi) = \sum_{i=1}^{N} P(\mathrm{O},q_0=i|\lambda')\log\pi_i$$

$$Q_{a_i}(\lambda',\mathrm{a}_i) = \sum_{j=1}^{N}\sum_{t=1}^{T} P(\mathrm{O},q_{t-1}=i,q_t=j|\lambda')\log a_{ij}$$

$$Q_{b_i}(\lambda',\mathrm{b}_i) = \sum_{t=1}^{T} P(\mathrm{O},q_t=i|\lambda')\ \log b_i(\mathrm{o}_t)$$

$Q(\lambda',\lambda)$ は 3 つの独立な項に分れているので、次の統計的制約に従って、それぞれの項を個別に最大化することで $Q(\lambda',\lambda)$ を $\lambda$ に関して最大化することができる。

$$\sum_{j=1}^{N}\pi_j = 1$$

$$\sum_{j=1}^{N} a_{ij} = 1,\quad \forall i$$

($b_i(\mathrm{o}_t=\mathrm{v}_k)=b_i(k)$) の離散分布に対して)

$$\sum_{k=1}^{K} b_i(k) = 1,\quad \forall i$$

それぞれの補助関数はすべて、



$$\sum_{j=1}^{N} w_j\log y_j$$

の形をしている。これは、$\sum_{j=1}^{N} y_j = 1, (y_j \geq 0)$ の制約が存在する $\{y_j\}_{j=1}^{N}$ の関数として、次の唯一の大域的最大点に至る。

$$y_j = \frac{w_j}{\sum_{i=1}^{N} w_i},\quad j = 1,2,\ldots,N$$

よって、最大化処理は再推定モデル $\overline{\lambda} = [\overline{\pi},\overline{A},\overline{B}]$ を次の点に導く。

$$\bar{\pi}_i = \frac{P(\mathrm{O},q_0=i|\lambda)}{P(\mathrm{O}|\lambda)}$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T} P(\mathrm{O},q_{t-1}=i,\ q_t=j|\lambda)}{\sum_{t=1}^{T} P(\mathrm{O},q_{t-1}=i|\lambda)}$$

$$\bar{b}_i(k) = \frac{\sum_{t=1}^{T} P(\mathrm{O},q_t=i|\lambda)\ \ \delta(\mathrm{o}_t,\mathrm{v}_k)}{\sum_{t=1}^{T} P(\mathrm{O},\ q_t=i|\lambda)}$$

ここで、

$$\delta(\mathrm{o}_t,\mathrm{v}_k) = 1\quad (\mathrm{o}_t=\mathrm{v}_k\text{のとき})$$
$$= 0\quad (\text{それ以外})$$

前向き変数 $\alpha_t(i) = P(\mathrm{o}_1,\mathrm{o}_2,\ldots,\mathrm{o}_t,\ q_t=i|\lambda)$ と後ろ向き変数 $\beta_t(i) = P(\mathrm{o}_{t+1},\ldots,\mathrm{o}_T|q_t=i,\lambda)$ の定義を用いれば、再推定変換式は次のように容易に計算できる。

$$P(\mathrm{O},q_t=i|\lambda) = \alpha_t(i)\,\beta_t(i)$$

$$P(\mathrm{O}|\lambda) = \sum_{i=1}^{N}\alpha_t(i)\,\beta_t(i) = \sum_{i=1}^{N}\alpha_T(i)$$

$$P(\mathrm{O},q_{t-1}=i,\ q_t=j|\lambda) = \alpha_{t-1}(i)\,a_{ij}b_j(\mathrm{o}_t)\,\beta_t(j)$$

よって、

$$\bar{\pi}_i = \frac{\alpha_0(i)\,\beta_0(i)}{\sum_{j=1}^{N}\alpha_T(j)} = \gamma_0(i)$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T}\alpha_{t-1}(i)\,a_{ij}b_j(\mathrm{o}_t)\,\beta_t(j)}{\sum_{t=1}^{T}\alpha_{t-1}(i)\,\beta_{t-1}(i)} = \frac{\sum_{t=1}^{T}\xi_{t-1}(i,j)}{\sum_{t=1}^{T}\gamma_{t-1}(i)}$$

$$\bar{b}_i(k) = \frac{\sum_{t=1}^{T}\alpha_t(i)\,\beta_t(i)\delta(\mathrm{o}_t,\mathrm{v}_k)}{\sum_{t=1}^{T}\alpha_t(i)\,\beta_t(i)} = \frac{\sum_{t=1,\ \text{s.t. } \mathrm{o}_t=\mathrm{v}_k}^{T}\gamma_t(i)}{\sum_{t=1}^{T}\gamma_t(i)}$$

これらは式 (6.40a)–(6.40c) で与えられた式である。

### 6.4.4 再推定手法に関する注意

上記の再推定式は、統計量に対する EM アルゴリズム [17] の実現法であると容易に解釈できる。つまり、E (期待値) ステップでは補助関数 $Q(\lambda',\lambda)$ を計算し (これは $\log\ P(\mathrm{O},\mathrm{q}\mid\lambda)$ の期待値である)、M (最大化) ステップでは、$\bar{\lambda}$ を得るために $\lambda$ に関する $Q(\lambda',\lambda)$ の最大化処理を行なう。よって Baum-Welch 再推定式は、この特別な問題に対しては基本的に EM ステップと等価である。

再推定手法の重要な特性は、HMM パラメータの統計的制約、すなわち、

$$\sum_{i=1}^{N}\bar{\pi}_i = 1 \tag{6.43a}$$

$$\sum_{j=1}^{N}\bar{a}_{ij} = 1,\quad 1\le i\le N \tag{6.43b}$$

$$\sum_{k=1}^{M}\bar{b}_j(k) = 1,\quad 1\le j\le N \tag{6.43c}$$

が繰り返し計算の中に自動的に組み込まれていることである。パラメータ推定問題を、(式 (6.43) の制約に従う) 制約された条件下での $P(\mathrm{O}|\lambda)$ の最適化問題と見なすと、$P$ を最大化するための変分法を使って解決手段を定式化できる (簡単のためこの節では、$P = P(\mathrm{O}|\lambda)$ という表記を用いる)。



ラグランジュ乗数を用いた通常のラグランジュ最適化法に基づけば、$P$ は次の条件を満たすときに最大になることが容易にわかる。

$$\pi_i = \frac{\pi_i\dfrac{\partial P}{\partial \pi_i}}{\sum_{k=1}^{N}\pi_k\dfrac{\partial P}{\partial \pi_k}} \tag{6.44a}$$

$$a_{ij} = \frac{a_{ij}\dfrac{\partial P}{\partial a_{ij}}}{\sum_{k=1}^{N}a_{ik}\dfrac{\partial P}{\partial a_{ik}}} \tag{6.44b}$$

$$b_j(k) = \frac{b_j(k)\dfrac{\partial P}{\partial b_j(k)}}{\sum_{\ell=1}^{M}b_j(\ell)\dfrac{\partial P}{\partial b_j(\ell)}} \tag{6.44c}$$

式 (6.44) を適切に解けば、各式の右辺は式 (6.40a)–(6.40c) の各式の右辺に等しいことがすぐにわかる。これは、再推定式が $P$ の臨界点において、まさに厳密に正しいことを示している。実際、式 (6.44) の形式は基本的に、左辺が再推定値で、右辺が現在の変数の値を使って計算される再推定式の形式になっている。

全体の問題は最適化問題として組み立てることができるので、モデルパラメータの "最適" 値を求めるために、通常の勾配法を使うことができることも最後に記しておく。そのような手法が試みられた結果、通常の再推定手法と同等な解が得られることが明らかになっている [18]。通常の勾配法を $P(\mathrm{O}|\lambda)$ の最大化に適用した場合に起こる大きな欠点の 1 つは、降下アルゴリズムが、勾配方向に取る小さなステップに大きく依存してしまうので、尤度が単調に増加しないことがしばしばあることである。Baum-Welch 再推定ではそれが式 (6.42) によって保証されている。

## 6.5 HMM の種類

HMM を分類する 1 つの方法に、マルコフ連鎖の遷移行列 $A$ の構造によって分類する方法がある。これまで我々は、モデルのすべての状態へ、他のすべての状態から (1 ステップで) 到達できるエルゴディック HMMや、全結合 HMMなどの特別な場合だけを考えてきた (厳密に言えば、エルゴ

ディックモデルは、すべての状態が、他のすべての状態から、有限であるが非周期的なステップ回数で到達できる特性がある)。 図 6.8(a) に $N = 4$ 状態モデルの例を示す。この種類のモデルは、$a_{ij}$ 係数がすべて正であるという特性がある。 よって、図 6.8(a) の例では次のような遷移行列が存在する。

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

ある応用においては (特に、この章のあとで議論するものにおいては)、標準的なエルゴディックモデルよりも他の種類の HMM の方が、観測される信号の性質をより良くモデル化できることがわかっている。 そのようなモデルの 1 つを図 6.8(b) に示す。 このモデルは、モデルに関連する背後の状態系列が、時間が経過するにつれ状態番号も大きくなる (または同じままでいる) 特性、つまり、系の状態が左から右に進むという特性があるので、left-right モデル、あるいは Bakis モデルと呼ばれる [11], [10]。明らかに left-right HMM は、時間とともに連続的に特性を変えていく信号、例えば音声などを容易にモデル化することのできる理想的な特性を備えている。 すべての left-right HMM の基礎的な特性には、次のような状態遷移係数の特性がある。

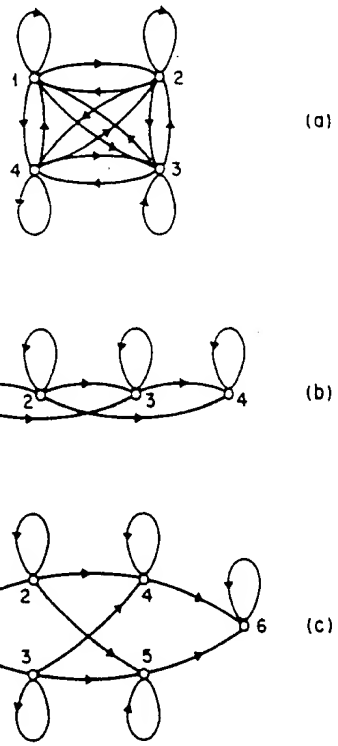



**図 6.8** 3 つの異なるタイプの HMM。(a) 4 状態エルゴディックモデル、(b) 4 状態 left-right モデル、(c) 6 状態平行パス left-right モデル



$$a_{ij} = 0, \quad j < i \tag{6.45}$$

つまり、現在の状態よりも小さい番号の状態へ遷移することが許されない。さらに、状態系列は状態 1 から始まり、状態 $N$ で終わらなければならないので、初期状態確率には次の特性がある。

$$\pi_i = \begin{cases} 0, & i \neq 1 \\ 1, & i = 1 \end{cases} \tag{6.46}$$

left-right モデルではしばしば、状態間を大きく遷移しないように、状態遷移係数にさらに制約を付け加えることがある。すなわち、次の形の制約がよく使われる。

$$a_{ij} = 0, \quad j > i + \Delta i \tag{6.47}$$

特に、図 6.8(b) の例では、$\Delta i$ の値は 2 である。つまり、2 状態以上の飛び越しは許されない。 よって、図 6.8(b) の例では状態遷移行列は次のようになる。

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix}$$

left-right モデルの最終状態の状態遷移係数が、以下のようになることは明らかであろう。

$$a_{NN} = 1 \tag{6.48a}$$

$$a_{Ni} = 0, \quad i < N \tag{6.48b}$$

上記の全結合モデルやleft-rightモデルの他に、たくさんの種類や組み合わせが考えられる。 例として、図6.8(c)に、2つの並列したleft-right HMMの"交差-連結"接続を示す。 厳密に言えば、このモデルはleft-rightモデルである(すべての$a_{ij}$の制約に従う)。しかし通常のleft-rightモデル(つまり、並列パスのないモデル)では表現できない自由度を備えている。

再推定手法は、モデルがleft-rightであっても、制約のある飛び越しが存在しても基本的に何ら影響を受けないことは明らかである。 なぜなら、初めにゼロに設定された遷移確率などのHMMパラメータは、再推定を行なっている間ずっとゼロのままで変わらないからである(式(6.44)を参照)。

## 6.6 HMMにおける連続観測確率密度

これまでの議論はすべて、観測事象が有限個のアルファベットから選択された離散シンボルである場合のみを考えてきた。よって、モデルの各状態では離散確率密度のみを用いることができた[19-21]。 このアプローチの問題点は、少なくともある応用においては、観測事象がしばしば連続信号(またはベクトル)であることである。 そのような連続信号表現をベクトル量子化符号帳などを使って離散シンボル系列に変換することはできるが、連続信号を離散シンボルに変換する際に重大な劣化が生じる可能性がある。 よって、連続信号表現を直接モデル化できる連続観測確率密度を備えたHMMが使えればより有効であろう。 連続観測確率密度を使うには、モデルの確率密度関数(pdf)の形式にいくらかの制約を与える必要がある。これは、pdfのパラメータを首尾一貫した方法で再推定するために必要である。 再推定手法が定式化されている最も一般的なpdfの表現形式は、次のような有限個の混合分布である。

$$b_j(\mathbf{o}) = \sum_{k=1}^{M} c_{jk}\, \mathcal{N}[\mathbf{o}, \boldsymbol{\mu}_{jk}, \mathbf{U}_{jk}], \quad 1 \le j \le N \tag{6.49}$$

ここで、$\mathbf{o}$はモデル化される観測ベクトルであり、$c_{jk}$は状態$j$の$k$番目の混合分布に対する混合重み係数、$\mathcal{N}$は対数凹(log-concave)、または楕円型対称確率密度[18]である(例えばガウス分布)。 式(6.49)において$\mathcal{N}$は、状態$j$の$k$番目の混合分布要素で、平均ベクトル$\boldsymbol{\mu}_{jk}$と共分散行列$\mathbf{U}_{jk}$を有するガウス分布であると仮定しても一般性は失われない。 混合重み係数$c_{jk}$は、次の統計的制約を満たす。



$$\sum_{k=1}^{M} c_{jk} = 1, \quad 1 \le j \le N \tag{6.50a}$$

$$c_{jk} \ge 0, \quad 1 \le j \le N, \ 1 \le k \le M \tag{6.50b}$$

これらによって、pdfは適切に正規化される。すなわち、

$$\int_{-\infty}^{\infty} b_j(\mathbf{o}) d\mathbf{o} = 1, \quad 1 \le j \le N \tag{6.51}$$

式(6.49)のpdfを用いて、任意の有限の連続確率密度関数を任意の近さで近似することができる。 よって、広範囲の問題に適用できる。

混合確率密度を持つHMMの状態は、単一確率密度を持つ複数状態のモデルと等価であることが以下のように証明されている[21]。 状態$i$の$M$混合ガウス確率密度を考える。 混合重み係数の和は1なので、それぞれの重み係数を用いて、サブ状態$i_1$(遷移確率$c_{i1}$)、$i_2$(遷移確率$c_{i2}$)から$i_M$(遷移確率$c_{iM}$)まで、サブ状態への遷移係数の集合が定義できる。 各サブ状態$i_k$には、平均$\boldsymbol{\mu}_{ik}$、分散$\mathbf{U}_{ik}$の単一混合分布が存在する(図的解釈は図6.9を参照)。 各サブ状態から待機状態$i_0$へは確率1で遷移する。 これらサブ状態(各々は単一確率密度)の複合集合の分布は、1状態の複合した混合確率密度分布と数学的に等価である。

混合確率密度係数、つまり$c_{jk}$、$\boldsymbol{\mu}_{jk}$、$\mathbf{U}_{jk}$に対する再推定式は次のよう

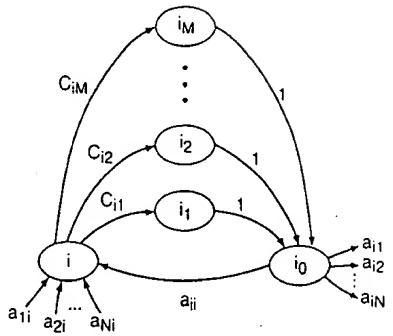



図 6.9 1状態混合確率密度分布と等価な複数状態単一確率密度分布(Juangらによる[21])

に示される。

$$\bar{c}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)}{\sum_{t=1}^{T}\sum_{k=1}^{M} \gamma_t(j,k)} \tag{6.52}$$

$$\bar{\mu}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k) \cdot \mathrm{o}_t}{\sum_{t=1}^{T} \gamma_t(j,k)} \tag{6.53}$$

$$\bar{\mathrm{U}}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k) \cdot (\mathrm{o}_t - \mu_{jk})(\mathrm{o}_t - \mu_{jk})'}{\sum_{t=1}^{T} \gamma_t(j,k)} \tag{6.54}$$

ここで、プライムはベクトルの転置を表し、$\gamma_t(j,k)$ は、$\mathrm{o}_t$ を考慮したときに、時刻 $t$ に状態 $j$ の $k$ 番目の混合要素に存在する確率である。すなわち、

$$\gamma_t(j,k) = \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}\right]\left[\frac{c_{jk}\,\mathcal{N}(\mathrm{o}_t, \mu_{jk}, \mathrm{U}_{jk})}{\sum_{m=1}^{M} c_{jm}\,\mathcal{N}(\mathrm{o}_t, \mu_{jm}, \mathrm{U}_{jm})}\right]$$

(離散確率密度分布や単一混合分布の場合は、$\gamma_t(j,k)$ 項は式 (6.26) の $\gamma_t(j)$ に一般化される。) $a_{ij}$ の再推定式は、離散観測確率密度で用いたものと同一である (すなわち、式 (6.40(b))。 式 (6.52)-(6.54) の解釈はとても簡単である。 $c_{jk}$ の再推定式は、系が状態 $j$ の $k$ 番目の混合要素に存在する回数の期待値と、系が状態 $j$ に存在する回数の期待値の比を示している。 同様に、平均ベクトル $\mu_{jk}$ に対する再推定式は、式 (6.52) の分子の項を観測ベクトルで重みづけしている。これによって、観測ベクトルが、$k$ 番目の混合要素によって説明される部分の期待値が得られる。 同様の解釈が、共分散行列 $\mathrm{U}_{jk}$ の再推定項に対しても言える。

## 6.7 自己回帰 HMM

音声処理に適用可能で、たいへん興味あるもう 1 つの HMM の種類に自己回帰 HMM[22-23] がある。 この種類の HMM は、観測ベクトルが自己回帰過程から生成される (もちろん、自己回帰確率密度は単に別の連続確率密度に過ぎない。しかし、あとで示すように、音声分析において自己回帰確率密度は重要であるので、ここではこの主題について 6.6 節とは別に詳しく述べる)。



よりわかりやすくするために、観測ベクトル $\mathrm{o} = (x_0, x_1, x_2, \ldots, x_{K-1})$ を考える。 要素 $x_i$ は単に音声波形のサンプルと考えることもできる。 o の要素は、次の関係を満たす自己回帰ガウス情報源から出力されたと仮定する。

$$x_k = -\sum_{i=1}^{p} a_i x_{k-i} + e_k \tag{6.55}$$

ここで、$e_k$, $(k = 0, 1, 2, \ldots, K-1)$ は平均値ゼロ、分散値 $\sigma_e^2$ のガウス、独立、同一分布不規則変数で、$a_i$, $(i = 1, 2, \ldots, p)$ は自己回帰係数、または予測係数である。 $K$ が大きい場合 [22, 23]、o の確率密度関数はおよそ次のようである。

$$f(\mathrm{o}) = (2\pi\sigma_e^2)^{-K/2} \exp\left\{-\frac{1}{2\sigma_e^2}\delta(\mathrm{o},\mathrm{a})\right\} \tag{6.56}$$

ここで、

$$\delta(\mathrm{o},\mathrm{a}) = r_a(0)r(0) + 2\sum_{i=1}^{p} r_a(i)r(i) \tag{6.57a}$$

$$\mathrm{a} = [1, a_1, a_2, \ldots, a_p]', \quad (a_0 = 1) \tag{6.57b}$$

$$r_a(i) = \sum_{n=0}^{p-i} a_n a_{n+i}, \qquad 1 \le i \le p \tag{6.57c}$$

$$r(i) = \sum_{n=0}^{K-i-1} x_n x_{n+i}, \quad 0 \le i \le p \tag{6.57d}$$

上記の式で、$r(i)$ は観測サンプルの自己相関で、$r_a(i)$ は自己回帰係数の自己相関である。 さらに、$\delta(\mathrm{o},\mathrm{a})$ は、a で定義される全零フィルタを用いて、データ $x_i$ を逆フィルタリングした結果の残差エネルギーの形式をしている (式 4.40–44 参照)。

4 章で議論したように、音声パターンの比較において、観測 o の信号レベルはしばしば、一般的なスペクトルの概形とは違った形で扱われる。 信号レベルをスペクトルの概形から分離する 1 つの方法は、ゲイン正規化を用いることである。つまり、o の代わりに $\hat{\mathrm{o}}$ を用いる。ここで、

$$\hat{\mathbf{o}} = \mathbf{o}/\sigma_{e\mathbf{o}} \tag{6.58}$$

$\sigma^2_{e\mathbf{o}}$ はサンプル当たりの最小線形予測残差エネルギーである (練習 6.5 で、与えられた観測 o に対し、$\sigma^2_{e\mathbf{o}} = (\sigma^2_e)_{\mathrm{ML}}$ であると示された)。 $\hat{\mathbf{o}}$ の要素 $\hat{x}_k = x_k/\sigma_{e\mathbf{o}}$ は依然、次の自己回帰関係を満たす。

$$\hat{x}_k = -\sum_{i=1}^{p} a_i \hat{x}_{k-i} + \hat{e}_k \tag{6.59}$$

ところが今度は、$\hat{e}_k$ の分散値が 1 である。 よって、平均値が 0、分散値が 1 のガウス、独立、同一分布系列より導出された a で定義される全極型システムの出力に対する確率密度関数は、データ次数 $K$ が十分に大きいならば以下のように書ける (正規化係数 $\sigma_{e\mathbf{o}}$ はもとの観測データ o に依存することに注意)。

$$f(\hat{\mathbf{o}}) = (2\pi)^{-K/2} \exp\left\{-\frac{1}{2}\delta(\hat{\mathbf{o}}, \mathbf{a})\right\} \tag{6.60}$$

このタイプの pdf はしばしば "ゲイン独立"pdf と呼ばれる。

ガウス自己回帰確率密度を HMM に用いる方法は簡単である。 次の混合確率密度を仮定する。

$$b_j(\mathbf{o}) = \sum_{k=1}^{M} c_{jk} b_{jk}(\mathbf{o}) \tag{6.61}$$

ここで、各 $b_{jk}(\mathbf{o})$ は自己回帰ベクトル $\mathbf{a}_{jk}$(または、等価的に自己相関ベクトル $\mathbf{r}_{a_{jk}}$) を用いて、式 (6.60) で定義される確率密度である。つまり、

$$b_{jk}(\mathbf{o}) = (2\pi)^{-K/2} \exp\left\{-\frac{1}{2}\delta(\mathbf{o}, \mathbf{a}_{jk})\right\} \tag{6.62}$$

状態 $j$ の $k$ 番目の混合要素に対する系列自己相関の再推定式がすでに導出されており [22, 23]、次の形を有する。

$$\bar{\mathbf{r}}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k) \cdot \mathbf{r}_t}{\sum_{t=1}^{T} \gamma_t(j,k)} \tag{6.63a}$$

ここで、$\mathbf{r}_t = [r_t(0), r_t(1), \ldots, r_t(p)]'$ は、式 (6.57d) で定義した $t$ 番目のフレームに対する自己相関ベクトルであり、$\gamma_t(j,k)$ は、時刻 $t$ で状態 $j$ に $k$ 番目の混合要素を用いて存在する確率として定義される。すなわち、



$$\gamma_t(j,k) = \left[\frac{\alpha_t(j)\,\beta_t(j)}{\sum_{j=1}^{N} \alpha_t(j)\,\beta_t(j)}\right]\left[\frac{c_{jk} b_{jk}(\mathbf{o}_t)}{\sum_{k=1}^{M} c_{jk} b_{jk}(\mathbf{o}_t)}\right] \tag{6.63b}$$

$\bar{\mathbf{r}}_{jk}$ は、観測系列中のフレームの正規化自己相関を、(出現確率によって) 重み付けて和を計算したものであることがわかる。 $\bar{\mathbf{r}}_{jk}$ から、 状態 $j$ の $k$ 番目の混合要素に関係する自己回帰係数ベクトル $\bar{\mathbf{a}}_{jk}$ を得るための、一連の正規式を解くことができる。 確率密度関数に必要な自己回帰係数の新しい自己相関ベクトルは、式 (6.57c) によって計算でき、これで再推定のためのループを閉じることができる。

**練習 6.4**

式 (6.56) の確率密度関数 (pdf)[22, 23] は、パラメータ $\sigma^2_e$ と a によって定義される。 与えられた観測ベクトル $\mathbf{o} = (x_0, x_1, \ldots, x_{K-1})$ を最もよく表現する $\sigma^2_e$ と a の最尤推定値を決定せよ。

**解答 6.4**

o の尤度関数を、$\sigma^2_e$ と a の関数として書くと次のようになる。

$$f(\mathbf{o}|\sigma^2_e, \mathbf{a}) = (2\pi\sigma^2_e)^{-K/2} \exp\left\{-\frac{1}{2\sigma^2_e}\,\delta(\mathbf{o}, \mathbf{a})\right\}$$

そして、その対数尤度関数は、

$$\log f(\mathbf{o}|\sigma^2_e, \mathbf{a}) = -\frac{K}{2}\log(2\pi\sigma^2_e) - \frac{\delta(\mathbf{o}, \mathbf{a})}{2\sigma^2_e}$$

よって、最尤 (ML) 推定値は、

$$\begin{aligned}(\mathbf{a})_{\mathrm{ML}} = \operatorname*{argmax}_{\mathbf{a}} \log f(\mathbf{o}|\sigma^2_e, \mathbf{a}) &= \operatorname*{argmin}_{\mathbf{a}} \delta(\mathbf{o}, \mathbf{a}) \\ &= \operatorname*{argmin}_{\mathbf{a}} (\mathbf{a}'\mathbf{R}\mathbf{a})\end{aligned}$$

ここで、式 (6.57d) で定義したように、$\mathbf{R} = [r_{ij}]$、$r_{ij} = r(|i-j|)$ である。 これは、LPC 分析における最尤推定と古典的な自己相関マッチング法 (予測残差最小法とも呼ばれる) の関係を表している (式 (4.8)–(4.10) を見よ)。さらに、容易に次式が導ける。

$$(\sigma^2_e)_{\mathrm{ML}} = \min_{\mathbf{a}} \delta(\mathbf{o}, \mathbf{a})/K$$

これから、次が導ける。

$$\max_{\sigma_e^2,\mathrm{a}} \log f(\mathrm{o}|\sigma_e^2,\mathrm{a}) = -\frac{K}{2}\log\left[2\pi(\sigma_e^2)_{\mathrm{ML}}\right] - \frac{K}{2}$$

**練習 6.5**

式 (6.56) の pdf は、式 (4.45) の板倉 - 斎藤歪み尺度に関係している。この関係を確立せよ。

**解答 6.5**

次の尤度差を考える。

$$\begin{aligned} L_d &= \left[\max_{\sigma_e^2,\mathrm{a}} \log f(\mathrm{o}|\sigma_e^2,\mathrm{a})\right] - \log f(\mathrm{o}|\sigma_e^2,\mathrm{a}) \\ &= -\frac{K}{2}\log[2\pi(\sigma_e^2)_{\mathrm{ML}}] - \frac{K}{2} + \frac{K}{2}\log(2\pi\sigma_e^2) + \frac{\delta(\mathrm{o},\mathrm{a})}{2\sigma_e^2} \\ &= \frac{K}{2}\left[\frac{1}{K\sigma_e^2}\delta(\mathrm{o},\mathrm{a}) + \log\sigma_e^2 - \log(\sigma_e^2)_{\mathrm{ML}} - 1\right] \end{aligned}$$

歪み尺度に関し、全極 (パワー) スペクトルを $\sigma_{e\mathrm{o}}^2/|A_{\mathrm{o}}(e^{j\omega})|^2$ と $\sigma_e^2/|A(e^{j\omega})|^2$ で表す。そして、それぞれに関係するパラメータ集合を $\{(\sigma_e^2)_{\mathrm{ML}}, (\mathrm{a})_{\mathrm{ML}}\}$ と $\{\sigma_e^2,\mathrm{a}\}$ で表す ($\sigma_{e\mathrm{o}}^2 = (\sigma_e^2)_{\mathrm{ML}}$)。 すると、対数尤度差 $L_d$ の中の括弧でくくられた項は、式 (4.45) によれば、単に $d_{\mathrm{IS}}(\sigma_{e\mathrm{o}}^2/|A_{\mathrm{o}}|^2,\ \sigma_e^2/|A|^2)$ となる。なぜなら、自己相関マッチングと式 (6.57a) により、

$$\sigma_{e\mathrm{o}}^2\int_{-\pi}^{\pi}\frac{|A(e^{j\omega})|^2}{|A_{\mathrm{o}}(e^{j\omega})|^2}\frac{d\omega}{2\pi} = \frac{1}{K}\delta(\mathrm{o},\mathrm{a})$$

であるからである (我々は $\sigma_e^2$ (と $\sigma_{e\mathrm{o}}^2$) をサンプル分散として定義したことに注意せよ。もし、サンプル分散の代わりに全フレームの予測残差の分散を使ったならば、係数 $K$ は上記の式から消える)。よって、

$$\begin{aligned} &f(\mathrm{o}|\sigma_e^2,\mathrm{a}) \\ &= (2\pi\sigma_e^2)^{-K/2}\ \exp\left\{-\frac{K}{2}\left[d_{\mathrm{IS}}\left(\frac{\sigma_{e\mathrm{o}}^2}{|A_{\mathrm{o}}|^2},\frac{\sigma_e^2}{|A|^2}\right) + \log\sigma_{e\mathrm{o}}^2 - \log\sigma_e^2 + 1\right]\right\} \\ &= G_1(\sigma_{e\mathrm{o}}^2,\sigma_e^2)\ \exp\left\{-\frac{K}{2}d_{\mathrm{IS}}\left(\frac{\sigma_{e\mathrm{o}}^2}{|A_{\mathrm{o}}|^2},\frac{\sigma_e^2}{|A|^2}\right)\right\} \end{aligned}$$

ここで、$G_1(\sigma_{e\mathrm{o}}^2,\sigma_e^2)$ はゲイン項のみを含む。



**練習 6.6**

式 (6.60) の pdf は、式 (4.53) の尤度比歪み尺度と関係がある。この関係を確立せよ。

**解答 6.6**

式 (6.56) の場合と同様に、

$$\begin{aligned} \frac{1}{K}\delta(\hat{\mathrm{o}},\mathrm{a}) &= \int_{-\pi}^{\pi}\frac{|A(e^{j\omega})|^2}{|A_{\mathrm{o}}(e^{j\omega})|^2}\frac{d\omega}{2\pi} \\ &= d_{\mathrm{LR}}\left(\frac{1}{|A_{\mathrm{o}}|^2},\frac{1}{|A|^2}\right) + 1 \end{aligned}$$

よって、

$$\begin{aligned} f(\hat{\mathrm{o}}|\mathrm{a}) &= (2\pi)^{-K/2}\ \exp\left\{-\frac{K}{2}\left[d_{\mathrm{LR}}\left(\frac{1}{|A_{\mathrm{o}}|^2},\frac{1}{|A|^2}\right) + 1\right]\right\} \\ &= G_2\ \exp\left\{-\frac{K}{2}\,d_{\mathrm{LR}}\left(\frac{1}{|A_{\mathrm{o}}|^2},\frac{1}{|A|^2}\right)\right\} \end{aligned}$$

pdf が歪み尺度 ($d_{\mathrm{IS}}$ または $d_{\mathrm{LR}}$) によって表現されている場合、指数項はデータの次数を表す係数 $K$ を含んでいることに注意せよ。実際には、この係数 $K$ は有効フレーム長 $\hat{K}$ に置き換えられる。 $\hat{K}$ は、連続するデータフレームの正味のシフト量である 。 仮に、連続するデータフレーム (ベクトル) が 2/3 の重なりをもっているならば、有効フレーム長 $\hat{K} = K/3$ が適切である。これにより、スペクトルパラメータ a に関する特徴の変化速度は、もとの波形の標本化速度を保つことができる。

## 6.8 HMM 構造の変形 — ナル遷移と状態の結び

この章ではこれまで、観測事象がモデルの状態に対応づけられる HMM を考えてきた。 観測事象がアークと対応づけられるモデルも考えることができ、この種類の HMM は IBM の連続音声認識装置に広く使われてきた [13]。 この種類のモデルは、何も出力せずに遷移する、すなわち、観測事象を生成せずに、ある状態から他の状態へ移動できるようにする場合に便利であることがわかっている [13]。 このような遷移をナル遷移と呼び、ナ

ル出力を表すためのシンボル $\phi$ とともに破線で示す。

図 6.10 にナルアークを使って成功した 3 つの (音声処理タスクの) 例を示す。 (a) は、状態が多数ある HMM (left-right モデル) でありながら、あらゆる状態間の遷移を省略できるモデルである。 よって、状態 1 から始まり状態 $N$ で終わるパスを依然として考慮しながら、たった 1 つの観測事象だけが生成されるということが可能である。

図 6.10(b) は、言語的な単位モデルを用いて単語を表現した有限状態ネットワーク (finite-state netword: FSN) の例である (つまり、各アークに付けられた音が、それ自身 HMM である)。このモデルでは、ナル遷移によって単語の発音の変化 (すなわち、シンボルの欠落) を簡潔かつ効率的に記述できる。

最後に、図 6.10(c) の FSN は、ナル遷移を文法ネットワークに挿入することによって、任意の長さの単語 (数字) 系列が、比較的簡単なネットワークでいかに生成できるかを示したものである。 図 6.10(c) の例では、各数字が生成された後にナル遷移で初期状態に戻ることにより、任意の長さの任意の数字系列を生成するネットワークができあがっている。

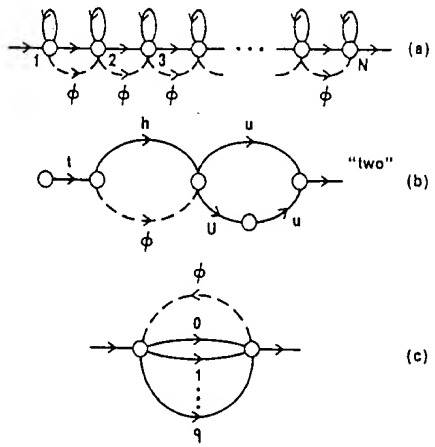



**図 6.10** ナル遷移のあるネットワークの例。 (a) left-right モデル、 (b) 有限状態ネットワーク、 (c) 文法ネットワーク



もう 1 つ興味ある HMM の構造の変形は、パラメータの結びの概念である [13]。 これは基本的に、異なる状態間で HMM パラメータを共有する考え方である。 これにより、独立なモデルパラメータが数多く削減され、パラメータ推定がいくらか簡単になり、より信頼度が増す場合がある。 パラメータの結びは、例えば、2 つまたはそれ以上の状態間で観測確率密度が同じであるとわかっているときに使われる。 これは音声を表現する場合にしばしば起こる。 特にこの手法は、数多くのモデルパラメータを信頼度高く推定するだけの、十分な学習データがない場合に有効である。 このような場合、パラメータ数 (つまり、モデルの大きさ) を削減するようにモデルパラメータを結びにすることが適切である。これにより、パラメータ推定がいくらか簡単になる。 この方法についてはこの章のあとで議論する。

## 6.9 HMM への明示的状態継続時間長確率密度の組み込み

我々は以前、式 (6.5) によって、自己遷移係数が $a_{ii}$ である状態 $i$ の固有の継続時間長確率 $p_i(d)$ が次のようになることを示した。

$$\begin{aligned} p_i(d) &= (a_{ii})^{d-1}(1-a_{ii}) \\ &= \text{状態 } i \text{において観測事象が } d \text{ 個連続する確率} \end{aligned} \tag{6.64}$$

自然界にある多くの信号に対し、この指数的状態継続時間長確率密度は適切でない。 そこで、継続時間長確率密度を明示的に、ある解析的な形でモデル化したい (状態継続時間長モデルに関する広範囲の論述は、IDA の Fergusonの書 [14] に見ることができ、ここで示す題材の基礎となっている。他の重要な参考文献は [24][25] である。)。 図 6.11 は、モデルの状態の対 $i$、$j$ に対し、明示的な状態継続時間長確率密度を用いた HMM と用いない HMM の違いを示している。 (a) では、各状態は、それぞれの自己遷移係数 $a_{ii}$、$a_{jj}$ をもとにした指数的な状態継続時間長確率密度を有している。 (b) では自己遷移係数はゼロに設定されており、明示的な状態継続時間長確率密度が指定されている。 (b) の場合、観測事象が (状態継続時間長確率密度に示される) 適切な数だけ状態内で発生した場合にのみ状態が遷移する。このようなモデルをセミマルコフモデル(semi-Markov model) と呼ぶ。

図 6.11(b) の簡単なモデルに基づいた場合、可変継続時間長 HMM の観測事象の系列は以下のようになる。

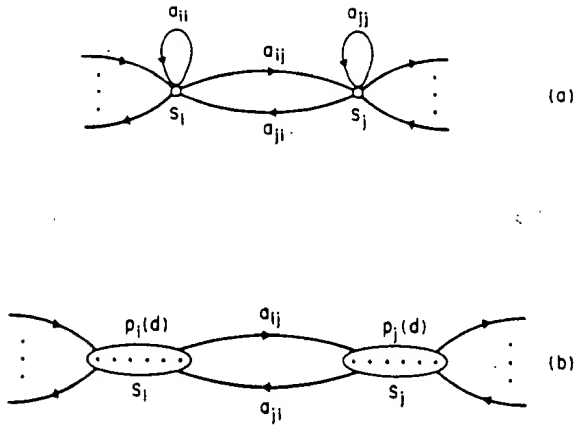



**図 6.11** (a) 指数的状態継続時間長確率密度のある通常の HMM と (b) その状態自身に戻る自己遷移がなく、指定された状態継続時間長確率密度のある可変継続時間長 HMM の一般的な状態間結合

1) 初期状態 $q_1 = i$ が初期状態分布 $\pi_i$ に従って選ばれる。

2) 継続時間長 $d_1$ が状態継続時間長確率密度 $p_{q_1}(d_1)$ に従って選ばれる (便宜上、または実現を容易にするために、状態継続長確率密度 $p_q(d)$ は最大継続長 $D$ で打ち切る)。

3) 観測事象 $\mathrm{o}_1\,\mathrm{o}_2 \ldots \mathrm{o}_{d_1}$ が同時確率密度 $b_{q_1}(\mathrm{o}_1\,\mathrm{o}_2 \ldots \mathrm{o}_{d_1})$ に従って選ばれる。一般に、各状態中の観測事象は独立と仮定するので、$b_{q_1}(\mathrm{o}_1\,\mathrm{o}_2 \ldots \mathrm{o}_{d_1}) = \Pi_{t=1}^{d_1} b_{q_1}(\mathrm{o}_t)$ となる。

4) 次の状態、$q_2 = j$ が状態遷移確率 $a_{q_1 q_2}$ によって選ばれる。ただし、$a_{q_1 q_1} = 0$ の制約がある。つまり、同じ状態に戻ってくる遷移はない (状態 $q_1$ ではちょうど $d_1$ 個の観測事象が発生することを仮定しているので、明らかにこの制約が必要である)。

可変継続時間長 HMM の $p_i(d)$ に、式 (6.64) の指数確率密度を設定すれば、通常の HMM と等価になることは容易に理解できるだろう。

上記の定式化を使って、$P(\mathbf{O}|\lambda)$ の計算や、すべてのモデルパラメータの再推定ができるようにするためには、6.4.3 節の式をいくつか変更しなければならない。ここで我々は、第 1 状態が時刻 $t = 1$ から始まり、最終状態が $t = T$ で終わると仮定する。そして前向き変数 $\alpha_t(i)$ を次のように定義する。



$$\alpha_t(i) = P(\mathrm{o}_1\,\mathrm{o}_2 \ldots \mathrm{o}_t,\ \text{時刻 } t \text{ に状態 } i \text{ にとどまることをやめる}|\lambda) \tag{6.65}$$

最初の $t$ 回の観測で合計 $r$ 個の状態を訪れたとする。それらの状態を $q_1$, $q_2, \ldots, q_r$ と表し、それぞれの状態の継続時間長を $d_1$, $d_2, \ldots, d_r$ とする。式 (6.65) の制約から、

$$q_r = i \tag{6.66a}$$

$$\sum_{s=1}^{r} d_s = t \tag{6.66b}$$

となる。よって、式 (6.65) は次のように書ける。

$$\begin{aligned}\alpha_t(i) = \sum_q \sum_d & \pi_{q_1} \cdot p_{q_1}(d_1) \cdot P(\mathrm{o}_1\,\mathrm{o}_2 \ldots \mathrm{o}_{d_1}|q_1) \\ & \cdot a_{q_1 q_2} p_{q_2}(d_2) P(\mathrm{o}_{d_1+1} \ldots \mathrm{o}_{d_1+d_2}|q_2) \ldots \\ & \cdot a_{q_{r-1} q_r} p_{q_r}(d_r) P(\mathrm{o}_{d_1+d_2+\ldots+d_{r-1}+1} \ldots \mathrm{o}_t|q_r)\end{aligned} \tag{6.67}$$

ここでは、すべての状態 $q$ と、可能なすべての状態継続時間長 $d$ に対して総和がとられる。帰納的に、$\alpha_t(j)$ は次のように書ける。

$$\alpha_t(j) = \sum_{i=1}^{N} \sum_{d=1}^{D} \alpha_{t-d}(i)\, a_{ij} p_j(d) \prod_{s=t-d+1}^{t} b_j(\mathrm{o}_s) \tag{6.68}$$

ここで、$D$ はすべての状態中の最大継続時間長である。$\alpha_t(j)$ の計算を初期化するために以下の計算を実行する。

$$\alpha_1(i) = \pi_i p_i(1) \cdot b_i(\mathrm{o}_1) \tag{6.69a}$$

$$\alpha_2(i) = \pi_i p_i(2) \prod_{s=1}^{2} b_i(\mathrm{o}_s) + \sum_{\substack{j=1 \\ j \neq i}}^{N} \alpha_1(j)\, a_{ji} p_i(1)\, b_i(\mathrm{o}_2) \tag{6.69b}$$

$$\alpha_3(i) = \pi_i p_i(3) \prod_{s=1}^{3} b_i(\mathrm{o}_s) + \sum_{d=1}^{2} \sum_{\substack{j=1 \\ j \neq i}}^{N} \alpha_{3-d}(j)\, a_{ji} p_i(d) \cdot \prod_{s=4-d}^{3} b_i(\mathrm{o}_s) \tag{6.69c}$$

これらを、$\alpha_D(i)$ が計算されるまで同様に行なう。こうして、すべての $t < D$ に対し式 (6.68) を用いることができる。モデル $\lambda$ が与えられたときの $\mathbf{O}$ の確率は、通常の HMM で計算していたのと同様、$\alpha$ を用いて以下のように書けるのは明らかであろう。

$$P(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{6.70}$$

可変継続時間長 HMM のすべての変数に対して再推定式を与えるために、3 つの前向き後ろ向き変数を定義する必要がある。すなわち、

$$\alpha_t^*(i) = P(\mathbf{o}_1\,\mathbf{o}_2 \ldots \mathbf{o}_t, \text{ 時刻 } t+1\text{に状態 } i \text{ にとどまり始める}|\lambda) \tag{6.71}$$

$$\beta_t(i) = P(\mathbf{o}_{t+1} \ldots \mathbf{o}_T|\text{ 時刻 } t \text{ に状態 } i \text{ にとどまることをやめる}, \lambda) \tag{6.72}$$

$$\beta_t^*(i) = P(\mathbf{o}_{t+1} \ldots \mathbf{o}_T|\text{ 時刻 } t+1\text{に状態 } i \text{ にとどまり始める}, \lambda) \tag{6.73}$$

$\alpha, \alpha^*, \beta, \beta^*$ の関係は以下のようである。

$$\alpha_t^*(j) = \sum_{i=1}^{N} \alpha_t(i)\, a_{ij} \tag{6.74}$$

$$\alpha_t(i) = \sum_{d=1}^{D} \alpha_{t-d}^*(i) p_i(d) \prod_{s=t-d+1}^{t} b_i(\mathbf{o}_s) \tag{6.75}$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} \beta_t^*(j) \tag{6.76}$$

$$\beta_t^*(i) = \sum_{d=1}^{D} \beta_{t+d}(i) p_i(d) \prod_{s=t+1}^{t+d} b_i(\mathbf{o}_s) \tag{6.77}$$

これらの関係と定義に基づくと、離散型観測事象をともなう可変継続時間長 HMM の再推定式は以下のようになる。

$$\bar{\pi}_i = \frac{\pi_i \beta_0^*(i)}{P(\mathbf{O}|\lambda)} \tag{6.78}$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T} \alpha_t(i)\, a_{ij} \beta_t^*(j)}{\sum_{j=1}^{N} \sum_{t=1}^{T} \alpha_t(i)\, a_{ij} \beta_t^*(j)} \tag{6.79}$$



$$\bar{b}_i(k) = \frac{\sum_{\substack{t=1 \\ \text{s.t. } \mathbf{o}_t = \mathbf{v}_k}}^{T} \left[ \sum_{\tau<t} \alpha_\tau^*(i) \cdot \beta_\tau^*(i) - \sum_{\tau<t} \alpha_\tau(i)\, \beta_\tau(i) \right]}{\sum_{k=1}^{M} \sum_{\substack{t=1 \\ \text{s.t. } \mathbf{o}_t = \mathbf{v}_k}}^{T} \left[ \sum_{\tau<t} \alpha_\tau^*(i) \cdot \beta_\tau^*(i) - \sum_{\tau<t} \alpha_\tau(i)\, \beta_\tau(i) \right]} \tag{6.80}$$

$$\bar{p}_i(d) = \frac{\sum_{t=1}^{T} \alpha_t^*(i) p_i(d)\, \beta_{t+d}(i) \prod_{s=t+1}^{t+d} b_i(\mathbf{o}_s)}{\sum_{d=1}^{D} \sum_{t=1}^{T} \alpha_t^*(i) p_i(d)\, \beta_{t+d}(i) \prod_{s=t+1}^{t+d} b_i(\mathbf{o}_s)} \tag{6.81}$$

再推定式の解釈は以下の通りである。 $\bar{\pi}_i$ に対する式は、$\mathbf{O}$ が与えられたとき、状態 $i$ が初期状態であった確率である。 $\bar{a}_{ij}$ に対する式は、時刻 $t$ で、ある状態を終了する $\alpha$ 項、時刻 $t+1$ で新しい状態が始まる $\beta$ 項が連結されているという条件を使うこと以外は、通常の HMM とほとんど同じである。 $\bar{b}_i(k)$ に対する式は、状態 $i$ における観測事象 $\mathbf{o}_t = \mathbf{v}_k$ の発生回数の期待値を、状態 $i$ におけるすべての観測事象の発生回数の期待値で正規化している。 最後に、$\bar{p}_i(d)$ の再推定式は、状態 $i$ が任意の継続時間長で発生した回数の期待値の比になっている。

いくつかの問題において、明示的な状態継続時間長確率密度を使った場合に、モデルの質が大幅に改善されることから、状態継続時間長確率密度を導入することが重要であることがわかる。 しかし、この節で議論した可変継続時間長モデルを使うことにはいくつかの欠点がある。 1 つは、可変継続時間長モデルを使うと計算量的な負荷が劇的に増加することである。式 (6.68)-(6.69) の前向き変数$\alpha_i(i)$ の定義と初期化条件から、およそ $D$ 倍の記憶容量と $D^2/2$ 倍の計算量が必要であることがわかる。 $D$ が 25 だとすると (これは多くの音声処理で実際的な値である)、計算量は 300 倍になる。 可変継続時間長モデルにおけるその他の問題点は、状態ごとに推定すべきパラメータの数が、通常の HMM パラメータに加えて ($D$ 個) 大きくなることである。 さらに、学習セット中に存在する $T$ 個の観測事象に対し、状態遷移回数は概して少なく、$p_i(d)$ を推定するためのデータ量が、通常の HMM で使われる場合よりもずっと少ない。 よって、可変継続時間長 HMM では、通常の HMM よりも再推定問題が難しくなる。

これらの問題のいくつかを軽減する 1 つの案は、上記のノンパラメトリックな $p_i(d)$ を使う代わりに、パラメトリックな状態継続時間長確率密度を使うことである [23-24]。 特に、この案は、$\mu_i$ と $\sigma_i^2$ のパラメータを持つ

ガウス関数族

$$p_i(d) = \mathcal{N}(d, \mu_i, \sigma_i^2) \tag{6.82}$$

や、$v_i$ と $\eta_i$ のパラメータ、平均値 $v_i\eta_i^{-1}$、分散 $v_i\eta_i^{-2}$ を持つガンマ関数族

$$p_i(d) = \frac{\eta_i^{v_i} d^{v_i-1} e^{-\eta_i d}}{\Gamma(v_i)} \tag{6.83}$$

を用いることを含む。$\eta_i, v_i$ の再推定式が導出され、良い結果を得ている。これまでに成功している他の方法としては、継続時間長の適当な範囲に対して、継続時間長が一様分布であると仮定して、パスを制約したビタービ復号法を使うものがある。

## 6.10 最適化基準 — ML、MMI、およびMDI

通常のML設計基準は、学習観測系列 O を用いて

$$\lambda_{\mathrm{ML}} = \underset{\lambda}{\operatorname{argmax}}\ P(\mathbf{O}|\lambda) \tag{6.84}$$

を満たすモデルパラメータ集合 $\lambda$ を導出することである。これまで議論してきた再推定アルゴリズムは、この最適化問題に対する解を与えるものである。

しかし、いくつかの懸念から、別の設計基準を考える必要がある場合がある[26-28]。HMMのような統計的モデル化手法の基本的な原理は、モデルパラメータを慎重にかつ正しく選択した場合に、信号や観測系列を適切にモデル化できるということである。この原理の問題点は、仮定されたモデル(ここではHMM)が、観測された信号をモデル化するのに不適切であるために、パラメータがいかに慎重に選ばれようとも、モデルの精密度に限界が生じる場合があることである。この状況はしばしば"モデルミスマッチ"と言われている。ここで最初に議論する別の最適化基準は、観測信号をより精密にモデル化するために、モデルミスマッチの問題を解決するものである。

いま、観測された信号 $\mathbf{O} = (\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T)$ が制約系列 $\mathcal{R} = (\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_T)$ に関係している場合を考える。例えば、$\mathbf{R}_t$ は観測事象 $\mathbf{o}_t$ を特徴づける自己相関行列のようなものである。このとき、明らかに O は、制約系列 $\mathcal{R}$ の条件を満たす、数えきれない程多数存在する観測系列の内の1つである。さらに、観測系列の確率分布の観点から、$\mathcal{R}$ も満足するような分布の集合が存在する。この集合を $\Omega(\mathcal{R})$ と表す。最小判別情報量(minimum discrimination information: MDI)は、与えられた制約 $\mathcal{R}$ のもとで、2つの確率尺度(ここでは、その内の1つはHMMの形式である)の近さを表す尺度であり、以下のように定義される。



$$\nu(\mathcal{R}, P_\lambda) \triangleq \inf_{Q \in \Omega(\mathcal{R})} I(Q : P_\lambda) \tag{6.85}$$

ここで、

$$I(Q : P_\lambda) = \int q(\mathbf{O}) \log \frac{q(\mathbf{O})}{p(\mathbf{O}|\lambda)}\, d\mathbf{O} \tag{6.86}$$

は、分布 $Q$ と $P_\lambda$ の間の判別情報量である(2つの関数 $q(\cdot)$ と $p(\cdot|\lambda)$ は、それぞれ $Q$ と $P_\lambda$ に関係する確率密度関数である)。判別情報量は、与えられた学習観測集合に基づいて計算される。

MDI基準は、$\nu(\mathcal{R}, P_\lambda)$ を最小にするモデルパラメータ集合 $\lambda$ を選択するものである。MDIは、モデル $P(\mathbf{O}|\lambda)$ が集合 $\Omega(\mathcal{R})$ の要素にできるだけ近くなるように、モデルパラメータ集合 $\lambda$ を選択することであると解釈できる。モデルの近さは常に、与えられた観測事象によって評価された判別情報量の観点から測られるので、学習系列の特徴がパラメータ選択に大きな影響を及ぼす。判別性を強調するので、モデル推定はもはや仮定されたモデルの形式だけでは規定できない。しかし、MDI最適化問題はML最適化問題のように簡単ではなく、簡潔で頑健な実現化手段は知られていない。

HMMを種々の音声認識問題を解決するために使おうとする場合に、HMMの最適化基準について、もう1つ重要な関連事項が生じる。いま、各単語をパラメータ集合 $\lambda_v$, $(v = 1, 2, \ldots, V)$ のHMMで表現した $V$ 単語の語彙を認識することを考える。また、$P(v)$ は単語 $v$, $(v = 1, 2, \ldots, V)$ に対する事前確率であると仮定する。HMMの集合 $\Lambda = \{\lambda_v\}$ と事前確率を用いて、任意の観測系列 O に対し、次のような確率尺度を定義する。

$$P_\Lambda(\mathbf{O}) = \sum_{v=1}^{V} P(\mathbf{O}|\lambda_v, v)\, P(v) \tag{6.87}$$

($P(\mathbf{O}|\lambda_v, v)$ は、単語 $v$ で条件づけられた確率であることを示す。我々は時に、モデルパラメータ $\lambda_v$ を推定目的のための不規則変数として扱う必要性から含める。$\lambda_v$ が固定された場合、明らかに、$P(\mathbf{O}|\lambda_v, v)$ は $\lambda_v$ でパラメータ化された条件付き確率となる。) これらのモデルを学習する

ために(つまり、対応するモデルの最適パラメータを推定するために)、発声内容が既知である(ラベルづけした)単語を用いる。 ラベルづけした学習系列を $\mathbf{O}^v$ と表す。上付き文字の $v$ は、$\mathbf{O}^v$ が単語 $v$ を発声したものであることを示す。 式(6.84)の通常のML基準は、$\mathbf{O}^v$ を用いて次の式を満たすモデルパラメータ $\lambda_v$ を推定することである。

$$(\lambda_v)_{\mathrm{ML}} = \underset{\lambda}{\operatorname{argmax}}\ P(\mathbf{O}^v|\lambda)$$

それぞれのモデルは、対応したラベルの(複数の)学習観測系列を用いて個別に推定される。 しかし、結果として生成されるモデルは、必ずしも認識誤り率を最小にする最適解ではない。

各モデルの"判別力"(つまり、正しい単語モデルによって生成された観測系列と、他の単語によって生成された系列とを区別する能力)を最大にすることを目的とする別の設計基準に、最大相互情報量(maximum mutual information: MMI)基準がある。観測系列 $\mathbf{O}^v$ と単語 $v$ の $\Lambda = \{\lambda_v\}, (v = 1, 2, \ldots, V)$ を用いてパラメータ化された相互情報量は、次のように表される。

$$I_\Lambda(\mathbf{O}^v, v) = \log \frac{P(\mathbf{O}^v, v|\Lambda)}{P_\Lambda(\mathbf{O}^v)\ P(v)} \tag{6.88}$$

なぜなら、

$$P(\mathbf{O}^v, v|\Lambda)/P(v) = P(\mathbf{O}^v|\lambda_v)\ ,$$

$$I_\Lambda(\mathbf{O}^v, v) = \log\ P(\mathbf{O}^v|\lambda_v) - \log \sum_{w=1}^{V} P(\mathbf{O}^v|\lambda_w, w)\ P(w) \tag{6.89}$$

MMI基準は、相互情報量を最大にするモデル集合 $\Lambda$ 全体を見つけ出すことである。

$$(\Lambda)_{\mathrm{MMI}} = \max_{\Lambda} \left\{ \sum_{v=1}^{V} I_\Lambda(\mathbf{O}^v, v) \right\} \tag{6.90}$$

MMI基準はML基準とは明らかに異なる。 両者は共に最小相互エントロピーアプローチである。 MLアプローチでは、単語が与えられたときのデータの分布に対するHMMが実験的な分布とマッチする。 MMIアプローチでは、データが与えられたときの単語の分布に対するモデルが実験的な分布とマッチする。 この表現はMMIアプローチの利点をわかりやすく説明している。 MMIアプローチの最適化手段では、たとえラベルづけされたただ1つの学習系列 $\mathbf{O}^v$ が使われたとしても、モデルパラメータ集合 $\Lambda$ 全体に関係する。ML基準では尤度 $P(\mathbf{O}^v|\lambda_v)$ のみが取り扱われるが、



MMI基準では、尤度 $P(\mathbf{O}^v|\lambda_v)$ を"背景確率(probability background)" $P_\Lambda(\mathbf{O}^v)$ と比較し、その差を最大化しようとする。 しかし、$(\Lambda)_{\mathrm{MMI}}$ は $(\Lambda)_{\mathrm{ML}}$ を得るほど簡単ではない。 式(6.90)を解くためにはしばしば、勾配法のような一般的最適化手法を使う必要がある。 そのような最適化手法を実現する場合には、しばしば計算上の問題が生じる。

## 6.11 種々のHMMの比較

HMMに関して以下のような興味ある問題がある。 $\lambda_1$ と $\lambda_2$ の2つのHMMが与えられたとき、2つのモデルの類似度を測る適切な尺度は何か[29]。 例として、次の2つのモデルの場合を考えよう。

$$\lambda_1 = (A_1, B_1, \pi_1)\ \ \lambda_2 = (A_2, B_2, \pi_2)$$

ここで、

$$A_1 = \begin{bmatrix} p & 1-p \\ 1-p & p \end{bmatrix}\ \ B_1 = \begin{bmatrix} q & 1-q \\ 1-q & q \end{bmatrix}\ \ \pi_1 = [1/2\ \ 1/2]$$

$$A_2 = \begin{bmatrix} r & 1-r \\ 1-r & r \end{bmatrix}\ \ B_2 = \begin{bmatrix} s & 1-s \\ 1-s & s \end{bmatrix}\ \ \pi_2 = [1/2\ \ 1/2]$$

観測シンボルに対して、同じ統計的性質を持つという意味で(つまり、すべての $\mathbf{v}_k$ に対して $E[\mathbf{o}_t = \mathbf{v}_k|\lambda_1] = E[\mathbf{o}_t = \mathbf{v}_k|\lambda_2]$ という意味で)、$\lambda_1$ が $\lambda_2$ と等しくなるためには、次の関係が要求される。

$$pq + (1-p)(1-q) = rs + (1-r)(1-s)$$

または、$s$ について解いて、

$$s = \frac{p + q - 2pq}{1 - 2r}$$

(任意に選んで) $p = 0.6$, $q = 0.7$, $r = 0.2$ であるとすると、$s = 13/30 \approx 0.433$ を得る。 よって、2つのモデル $\lambda_1$ と $\lambda_2$ が見かけ上、非常に異なっていても(すなわち、$A_1$ は $A_2$ とは非常に異なり、$B_1$ も $B_2$ とは非常に異なる)、モデルが統計的には等しいということがあり得る。

2つのマルコフモデル $\lambda_1$ と $\lambda_2$ の距離尺度 $D(\lambda_1, \lambda_2)$ を次のように定義することにより、モデル距離(非類似度)の概念を一般化できる[29]。

$$D(\lambda_1, \lambda_2) = \frac{1}{T}\left[\log P(\mathbf{O}^{(2)}|\lambda_1) - \log P(\mathbf{O}^{(2)}|\lambda_2)\right] \tag{6.91}$$

ここで、$\mathbf{O}^{(2)} = (\mathbf{o}_1 \mathbf{o}_2 \mathbf{o}_3 \ldots \mathbf{o}_T)$ はモデル $\lambda_2$ によって生成された観測系列である。基本的に式 (6.91) は、モデル $\lambda_1$ が、モデル $\lambda_2$ によって生成された観測事象に適合する度合いと、モデル $\lambda_2$ が、$\lambda_2$ 自身によって生成された観測事象に適合する度合いとを比較した尺度である。相互エントロピー、ダイバージェンス、または判別情報量 [29] の観点から、式 (6.91) の解釈がいくつか存在する。

式 (6.91) の距離尺度の問題点の 1 つは、それが非対称形であることである。よって、この式の尺度の自然な形式は、対称化された形式、つまり、

$$D_s(\lambda_1, \lambda_2) = \frac{D(\lambda_1\lambda_2) + D(\lambda_2, \lambda_1)}{2} \tag{6.92}$$

である。

## 6.12 HMM の実現上の課題

前節までの議論では、主に HMM の理論と、いくつかのモデル形式の種類について取り上げた。この節では、実際にモデルを実現する上での問題点、例えばスケーリング、多数観測系列、初期パラメータ推定、データの欠落、モデルの大きさと種類の選択について取り上げる。これら実現上の問題点のいくつかに対しては、厳密な解析的解法を述べることができるが、他の問題点に対しては、HMM を研究してきた中で得られた経験のみを示すことしかできない。

### 6.12.1 スケーリング

HMM の再推定手法を実現する上で、なぜスケーリングが必要か [18,23] を理解するために、式 (6.18) の $\alpha_t(i)$ の定義を考えよう。$\alpha_t(i)$ は次に示すような多数の項の和から成ることがわかる。

$$\left(\prod_{s=1}^{t-1} a_{q_s q_{s+1}} \prod_{s=1}^{t} b_{q_s}(\mathbf{o}_s)\right)$$

ここで、$q_t = i$ で、$b$ は式 (6.8) で定義したような離散確率である。各 $a$, $b$ 項は 1 よりも小さいので (一般に 1 よりずっと小さい)、$t$ が大きくなり始めると (例えば 10、またはそれ以上)、$\alpha_t(i)$ の各項は指数的に 0 に向かい始める。十分大きい $t$ に対しては (例えば 100、またはそれ以上)、$\alpha_t(i)$ の計算のダイナミックレンジは、基本的にあらゆる計算機の精度の範囲を (たとえ倍精度であっても) 越えてしまう。よって、計算を実行するために唯一、適切な方法は、スケーリング処理を導入することである。



基本的なスケーリング処理としては、$i$ に独立な (つまり、$t$ のみに依存する) スケーリング係数を $\alpha_t(i)$ に掛けあわせ、スケーリングした $\alpha_t(i)$ が、$1 \le t \le T$ の間で計算機のダイナミックレンジ内に収まるようにすることである。同様のスケーリングを $\beta_t(i)$ 係数に対しても行ない (なぜならこれらも 0 に指数的に急速に近づく)、計算の最後にスケーリング係数を厳密に取り除く。

このスケーリング処理をもっと理解するために、状態遷移係数 $a_{ij}$ の再推定式を考える。再推定式 (Eq.(6.40b)) を前向き変数と後ろ向き変数を用いて直接書くと次の式を得る。

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \alpha_t(i)\, a_{ij} b_j(\mathbf{o}_{t+1})\, \beta_{t+1}(j)}{\sum_{t=1}^{T}\sum_{j=1}^{N} \alpha_t(i)\, a_{ij} b_j(\mathbf{o}_{t+1})\, \beta_{t+1}(j)} \tag{6.93}$$

$\alpha_t(i)$ の計算を考えよう。スケーリングしていない $\alpha$ を示すために $\alpha_t(i)$ を、スケーリングした (そして繰り返された) $\alpha$ を示すために $\hat{\alpha}_t(i)$ を、そしてスケーリング前の $\alpha$ の局所的な値を示すために $\widehat{\alpha}_t(i)$ を使う。あらかじめ $t = 1$ に対し、式 (6.19) に従い $\alpha_1(i)$ を計算し、$\widehat{\alpha}_1(i) = \alpha_1(i)$、$c_1 = \frac{1}{\sum_{i=1}^{N} \alpha_1(i)}$、$\hat{\alpha}_1(i) = c_1\alpha_1(i)$ とおく。初めにそれぞれの $t$、$(2 \le t \le T)$ に対し、式 (6.20) の帰納的な式に従い、以前スケーリングした $\hat{\alpha}_t(i)$ を使って $\widehat{\alpha}_t(i)$ を計算する。つまり、

$$\widehat{\alpha}_t(i) = \sum_{j=1}^{N} \hat{\alpha}_{t-1}(j)\, a_{ji}\, b_i(\mathbf{o}_t) \tag{6.94a}$$

スケーリング係数 $c_t$ を次のように決定する。

$$c_t = \frac{1}{\sum_{i=1}^{N} \widehat{\alpha}_t(i)} \tag{6.94b}$$

よって、

$$\hat{\alpha}_t(i) = c_t\, \widehat{\alpha}_t(i) \tag{6.94c}$$

式 (6.94a-c) から、スケーリングした $\hat{\alpha}_t(i)$ は $c_t\hat{\alpha}_t(i)$ と書ける。または次のようにも書ける。

$$\hat{\alpha}_t(i) = \frac{\sum_{j=1}^{N} \hat{\alpha}_{t-1}(j)\, a_{ji}\, b_i(\mathrm{o}_t)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \hat{\alpha}_{t-1}(j)\, a_{ji}\, b_i(\mathrm{o}_t)} \tag{6.95}$$

帰納法により、$\hat{\alpha}_{t-1}(j)$ は次のように書ける。

$$\hat{\alpha}_{t-1}(j) = \left(\prod_{\tau=1}^{t-1} c_\tau\right) \alpha_{t-1}(j) \tag{6.96a}$$

よって、$\hat{\alpha}_t(i)$ は、

$$\hat{\alpha}_t(i) = \frac{\sum_{j=1}^{N} \alpha_{t-1}(j) \left(\prod_{\tau=1}^{t-1} c_\tau\right) a_{ji}\, b_i(\mathrm{o}_t)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_{t-1}(j) \left(\prod_{\tau=1}^{t-1} c_\tau\right) a_{ji}\, b_i(\mathrm{o}_t)} = \frac{\alpha_t(i)}{\sum_{i=1}^{N} \alpha_t(i)} \tag{6.96b}$$

となる。つまり、各 $\alpha_t(i)$ はすべての状態の $\alpha_t(i)$ の和によって効果的にスケーリングされている。

次に $\beta_t(i)$ 項を後ろ向きの再帰演算により計算する。ここで唯一異なることは、各時刻 $t$ において、$\alpha$ 対して使ったのと同じスケーリング係数を $\beta$ に対しても用いることである。よって、スケーリングした $\beta$ は次のようになる。

$$\hat{\beta}_t(i) = c_t\beta_t(i) \tag{6.97}$$

各スケーリング係数は、$\alpha$ 項の大きさを効果的に 1 に回復する。また、$\alpha$ 項と $\beta$ 項の大きさは同程度であるので、$\alpha$ に使ったのと同じスケーリング係数を $\beta$ に対して用いることにより、計算量を適度な範囲に制限することができる。さらに、スケーリングした変数を用いると、式 (6.93) の再推定式は次のようになる。

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \hat{\alpha}_t(i)\, a_{ij}\, b_j(\mathrm{o}_{t+1})\, \hat{\beta}_{t+1}(j)}{\sum_{t=1}^{T-1}\sum_{j=1}^{N} \hat{\alpha}_t(i)\, a_{ij}\, b_j(\mathrm{o}_{t+1})\, \hat{\beta}_{t+1}(j)} \tag{6.98}$$

しかし、各 $\hat{\alpha}_t(i)$ は、



$$\hat{\alpha}_t(i) = \left[\prod_{s=1}^{t} c_s\right] \alpha_t(i) = C_t\alpha_t(i) \tag{6.99}$$

のように書くことができ、また、各 $\hat{\beta}_{t+1}(j)$ は、

$$\hat{\beta}_{t+1}(j) = \left[\prod_{s=t+1}^{T} c_s\right] \beta_{t+1}(j) = D_{t+1}\, \beta_{t+1}(j) \tag{6.100}$$

のように書けるので、式 (6.98) は、

$$\overline{a_{ij}} = \frac{\sum_{t=1}^{T-1} C_t\alpha_t(i)\, a_{ij}\, b_j(\mathrm{o}_{t+1}) D_{t+1}\, \beta_{t+1}(j)}{\sum_{t=1}^{T-1}\sum_{j=1}^{N} C_t\alpha_t(i) a_{ij}\, b_j(\mathrm{o}_{t+1})\, D_{t+1}\, \beta_{t+1}(j)} \tag{6.101}$$

のように書ける。最後に、$C_tD_{t+1}$ 項は、

$$C_tD_{t+1} = \prod_{s=1}^{t} c_s \prod_{s=t+1}^{T} c_s = \prod_{s=1}^{T} c_s = C_T \tag{6.102}$$

のようになって、$t$ に対して独立であることがわかる。よって、$C_tD_{t+1}$ 項は式 (6.101) の分母、分子から除外され、厳密な再推定式が実現される。

上記のスケーリング処理は、$\pi$ や $B$ 係数の再推定式にも同様に適用できることは明らかであろう。また、式 (6.95) のスケーリング処理は、すべての時刻 $t$ で適用する必要はなく、必要なとき、要求されたとき (例えばアンダーフローを避ける場合) のみ実行すればよいことも明らかであろう。ある時刻 $t$ でスケーリングが行なわれない場合は、スケーリング係数 $c_t$ は 1 に設定されるので、上で議論したすべての条件を満たす。

スケーリングによって唯一、実際に変更される HMM の手続きは、$P(\mathrm{O}|\lambda)$ の計算手続きである。$\hat{\alpha}_T(i)$ 項はすでにスケーリングされているので、単に総和をとることができない。しかし、次の特性を使うことができる。

$$\prod_{t=1}^{T} c_t \sum_{i=1}^{N} \alpha_T(i) = C_T \sum_{i=1}^{N} \alpha_T(i) = 1 \tag{6.103}$$

よって、

$$\prod_{t=1}^{T} c_t \cdot P(\mathrm{O}|\lambda) = 1 \tag{6.104}$$

または、

$$P(\mathbf{O}|\lambda) = \frac{1}{\prod_{t=1}^{T} c_t} \tag{6.105}$$

または、

$$\log\left[P(\mathbf{O}|\lambda)\right] = -\sum_{t=1}^{T} \log c_t \tag{6.106}$$

よって、$P$ ではなく、$P$ の対数値を計算することができる。なぜなら、$P$ は、何らかの形で計算機のダイナミックレンジを越えてしまうかもしれないからである。

最尤状態系列が得られるビタービアルゴリズムを用いる場合、ビタービの変形の実現法で議論したように、対数を使えばスケーリングは必要ないことを最後に記しておく。

### 6.12.2 複数観測系列

6.5 節で我々は、left-right モデルまたは Bakis モデルと呼ばれる HMM の形式について議論した。このモデルでは、状態は $t = 1$ で状態 1 から始まり $t = T$ で状態 $N$ に達するまで次々と進んでいく (図 6.8(b) のモデルを思い出して欲しい)。

我々は、left-right モデルの状態遷移行列や初期状態確率にどのような制約 (式 (6.45)–(6.48)) が加わるかについて既に議論した。しかし、left-right モデルの大きな問題点は、1 つの観測系列ではモデルの学習 (つまり、モデルパラメータの再推定) ができないことである。これは、モデル内の状態遷移の性質から、(次の状態に遷移するまでの間に) 状態に対して少数の観測事象しか割り当てられないためである。よって、すべてのモデルパラメータを信頼度高く推定するための十分なデータを得るためには、複数観測系列を使う必要がある [18]。

再推定手続きの変更は簡単で、以下のようである。$K$ 個の観測系列の集合を次のように表す。

$$\mathbf{O} = [\mathbf{O}^{(1)}, \mathbf{O}^{(2)}, \ldots, \mathbf{O}^{(K)}] \tag{6.107}$$

ここで、$\mathbf{O}^{(k)} = (\mathrm{o}_1^{(k)}\mathrm{o}_2^{(k)}\ldots\mathrm{o}_{T_k}^{(k)})$ は $k$ 番目の観測系列を示す。各観測系列は他のすべての観測系列に対して独立であると仮定する。我々の目的は、



$$P(\mathbf{O}|\lambda) = \prod_{k=1}^{K} P(\mathbf{O}^{(k)}|\lambda) \tag{6.108}$$

$$= \prod_{k=1}^{K} P_k \tag{6.109}$$

を最大にするようにモデル $\lambda$ のパラメータを調整することである。再推定式は、様々な事象の発生頻度に基づいているので、複数観測系列の再推定式は、個々の観測系列の発生頻度を加算するように変更することで実現できる。従って、$\bar{a}_{ij}$ と $\overline{b_j}(\ell)$ に対する変更された再推定式は、

$$\bar{a}_{ij} = \frac{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k-1} \alpha_t^k(i) a_{ij}\, b_j(\mathrm{o}_{t+1}^{(k)})\, \beta_{t+1}^k(j)}{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k-1} \alpha_t^k(i)\, \beta_t^k(i)} \tag{6.110}$$

$$\overline{b_j}(\ell) = \frac{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1 \text{ s.t. } \mathrm{O}_t=v_\ell}^{T_k-1} \alpha_t^k(i)\, \beta_t^k(i)}{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k-1} \alpha_t^k(i)\, \beta_t^k(i)} \tag{6.111}$$

となる。$\pi_i$ は $\pi_1 = 1$、$\pi_i = 0$、$(i \neq 1)$ なので再推定されない。

ここで、式 (6.110)–(6.111) に対して適切にスケーリングすることは簡単である。なぜなら、各観測系列はそれぞれのスケーリング係数を持っているからである。重要なのは、総和を取る前に各項からスケーリング係数の影響を取り除くことである。これは、スケーリングした変数を用いて再推定式を書くことによって実現できる。すなわち、

$$\bar{a}_{ij} = \frac{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k-1} \hat{\alpha}_t^k(i)\, a_{ij}\, b_j(\mathrm{o}_{t+1}^k)\, \hat{\beta}_{t+1}^k(j)}{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k-1} \hat{\alpha}_t^k(i) \hat{\beta}_t^k(i)} \tag{6.112}$$

このように各観測系列 $\mathbf{O}^{(k)}$ に対して、時間方向の総和をとる各項には同じスケーリング係数が現われることになるので厳密に取り除かれる。よって、スケーリングした変数 $\alpha$ と $\beta$ を用いても、結果的に、スケーリングしていない $\bar{a}_{ij}$ になる。同様な結果が $\overline{b_j}(\ell)$ 項に対しても得られる。

### 6.12.3 HMM パラメータの初期推定値

理論的には、再推定式は尤度関数の極大値に対応する HMM パラメータの値を与えるはずである。 問題は、その極大値が尤度関数の大域的最大値に等しくなるようにするためには、またはそれにできるだけ近くなるようにするためには、HMM パラメータの初期推定値をどう選べばよいかである。

基本的に、単純な、あるいは簡単な解はない。 しかし、$\pi$ や $A$ パラメータに対しては、(統計的で非ゼロの制約に従った) 不規則な、または一様な初期推定値を与えても、ほとんどすべての場合、これらのパラメータに対し、有益な再推定値を得るのに十分であることが経験的にわかっている。 しかしながら、$B$ パラメータに対しては、適切な初期推定値を用いることが、離散シンボルの場合には助けとなり、また連続分布の場合は (複数混合分布を扱う場合は) 必須であることが経験からわかっている。 そのような初期推定値は様々な方法で得られる。例えば、(a) (複数の) 観測系列を状態ごとに手作業でセグメンテーションして、状態内で観測事象の平均化処理をする、(b) 観測系列を最尤セグメンテーションして平均化処理をする、(c) クラスタリングを含むセグメンタル $K$- 平均 ($K$-means) セグメンテーションを行なう、などである。 このようなセグメンテーション手法については、この章のあとの方で議論する。

### 6.12.4 不十分な学習データの影響

再推定手法を用いた HMM の学習に伴うもう 1 つの問題点は、学習に用いる観測系列が必ず有限であるということである [30]。 よって、確率の低い事象 (例えば、状態内のシンボルの発生回数) に対しては、モデルパラメータを適切に推定するだけの十分な発生回数が得られないことが常に起こる。 例として、離散観測 HMM の場合を考えよう。 式 (6.40c) の $\bar{b}_j(k)$ の再推定変換式では、状態 $j$ に存在し、同時にシンボル $\mathbf{v}_k$ を観測すると期待される回数が必要であったことを思い出してほしい。 仮に、学習系列が非常に少なく、この事象 (すなわち、$q_t = j$ で $\mathbf{o}_t = \mathbf{v}_k$) が発生しないとしたら、$b_j(k) = 0$ となり、再推定後も 0 であり続けるということになる。 結果として生じるモデルは、($\mathbf{o}_t = \mathbf{v}_k$ で $q_t = j$) を実際に含むあらゆる観測系列に対して確率ゼロを与えるだろう。 明らかにそのようなおかしな結果は、学習セットが不十分なために $b_j(k) = 0$ となってしまった信頼できない推定値が原因である。



この問題に対する 1 つの解決法は、学習観測セットの大きさを大きくすることである。 これはしばしば実現不可能である。 第 2 の解決法は、モデルの大きさを小さくすることである (例えば、状態数、状態当たりのシンボル数など)。 これは常に可能であるが、与えられたモデルが使われる物理的な理由が存在する場合がしばしばあり、よって、モデルの大きさも変えられない場合もある。 第 3 の解決法は、学習データが限られたときでも、パラメータ推定の信頼度をいくらか補強できる、従来にない統計的推定アルゴリズムを探し出すことである。 削除補間法とパラメータしきい値法はその様な 2 つの手段である。 削除補間法は、より補強されたパラメータ推定法と考えられるので、これについては次の節で議論する。

学習データが不十分な場合の影響を取り扱う最も簡単な方法は、あらゆるモデルパラメータの推定値が、決められたレベル以下にならないように、モデルパラメータに対して特別なしきい値制限を設けることである [18]。 例えば、離散シンボルモデルに対し、次のような数値的な底上げを設定する。

$$b_j(k) = \begin{cases} b_j(k), & (b_j(k) \geq \delta_b \text{のとき}) \\ \delta_b, & (\text{それ以外}) \end{cases} \tag{6.113a}$$

また、連続分布モデルに対しては、

$$U_{jk}(r,\, \tau) = \begin{cases} U_{jk}(r,r), & (U_{jk}(r,\tau) \geq \delta_u \text{のとき}) \\ \delta_u, & (\text{それ以外}) \end{cases} \tag{6.113b}$$

再推定式において、数値的底上げをした場合、確率密度に要求される統計的制約に従うように、他のすべてのパラメータを再スケールする必要がある。 このような事後的な手法は、少数データの問題に対処するための実行上の措置と考えられているが、音声処理のいくつかの問題に対して適用され成功している。 パラメータしきい値法はベイズ統計の観点からも正当性がある。 事実、式 (6.113b) は、パラメータの事前分布 $P(U_{jk}(r,\tau))$

が一様分布で、$(U_{jk}(r,r))_{\min} = \delta_u$ であるという情報が事前にあるという条件のもとでの分散の最大事後確率 (MAP) 推定になっている [30]。(6.14 節参照)

### 6.12.5 モデルの選択

HMM を実現する際に生じる問題で、まだ残されているものには、モデルのタイプの選択 (エルゴディック、left-right、またはその他の形式)、モデルの大きさの選択 (状態数)、観測シンボルの選択 (離散または連続、単一分布または混合分布、観測パラメータの選択) がある。 不運にも、それらを選択するための簡単で、理論的に正しい手法はない。 これらは、モデル化する信号に依存して選択する必要がある。 このコメントをもって、隠れマルコフモデルの理論面の議論を終え、次に、このモデルがどの様に孤立単語認識の問題に応用されてきたかについての議論に進む。

## 6.13 モデル推定値の有効性の改善

この節では、音声認識のための、HMM のモデル推定値の有効性を高める 3 つの方法について議論する。 すなわち、(1) 削除補間法 (2) ベイズ適応法 (3) 誤り訂正学習法である。 初めの 2 つの方法は少数データの問題から動機づけられたものであり、最後の方法には、認識誤りを直接的に減らそうとする唯一の目標が存在する。

### 6.13.1 削除補間法

学習データ量が不十分なために、HMM パラメータを信頼度高く、かつ、頑健に決定できないことがしばしば起こる。 最尤基準に基づいた Baum-Welch 再推定法によって得られた HMM は、学習データを表現するのには適切であると思われるが、新しいデータに対する適合度は非常に悪いかもしれない。 モデルの信頼度を改善するためのパラメータ推定法の 1 つに "削除補間法" がある。

この方法の概念は、2 つの (またはそれ以上の) 個別にトレーニングされたモデル (そのうちの 1 つは他のモデルよりも信頼度高く学習されている) を結合することにある。 "異なる" 状態間に同一の統計的性質を共有させ、



モデル中のパラメータ数を効果的に削減する、状態の結び (tied) を使う場合にこのシナリオが起こり得る。 同じデータ量で学習した場合、状態の結びがあるモデルは、状態の結びがないモデルよりも頑健であることが多い。 しかし、学習データが十分にある場合は、状態の結びがあるモデルは、状態の結びがないモデルよりも精密度が低い。 したがって、より精密度が高いと思われるモデルが実際には信頼できない場合に、2 つのモデルを結合するという考え方によって、そのモデルをより信頼度の高いモデルに変えることができる。 (より頑健な) 文脈独立音素モデルと (より精密な) 文脈依存音素モデルを大語彙認識に使う場合にも、同様のシナリオが当てはまる (8 章を参照)。

2 つのモデルが、パラメータ集合 $\lambda = (A, B, \pi)$ と $\lambda' = (A', B', \pi')$ でそれぞれ定義されているとする。 補間したモデル $\tilde{\lambda} = (\tilde{A}, \tilde{B}, \tilde{\pi})$ は次のようにして得られる。

$$\tilde{\lambda} = \epsilon\lambda + (1-\epsilon)\lambda' \tag{6.114}$$

ここで、$\epsilon$ は (観測事象を、より詳細に表現している) "十分な" モデルに対するパラメータの重みを表し、$(1-\epsilon)$ は小規模だが信頼度の高いモデルに対するパラメータの重みを表す。 問題は、$\epsilon$ の最適値をいかに決定するかであり、この値は学習データ量と強い関係がある。 このことは容易に理解できる。なぜなら、学習データ量が多くなるにつれ、$\lambda$ はより信頼度が増し、$\epsilon$ は 1.0 に近づくと思われるからである。 同様に、少量の学習データに対しては、$\lambda$ の信頼度は低下し、より信頼度の高いモデル $\lambda'$ になるように、$\epsilon$ は 0.0 に近づくと思われる。

$\epsilon$ の最適値を決定する方法が Jelinek と Mercer によって提案された [30]。彼らは、式 (6.114) を図 6.12 に示すタイプの拡張 HMM と解釈して、最適な $\epsilon$ を、前向き後ろ向きアルゴリズムを使って推定できることを示した。図 6.12a は、状態 S に関係する状態遷移構造の一部分を示している。 式 (6.114) の補間モデルは、図 6.12b のように 1 状態を 3 状態に置き換えた拡張 HMM として解釈することができる。 拡張した状態 $\tilde{S}$ から $S$ および $S'$ へのナル遷移には、遷移確率 $\epsilon$ と $1-\epsilon$ がそれぞれ付与される。 $S$ から出ていく遷移は $\lambda$ の遷移で特徴づけられ、$S'$ から出ていく遷移は $\lambda'$ の遷移で関連づけられる。

この拡張 HMM の解釈により、パラメータ $\epsilon$ が、通常の前向き後ろ向きアルゴリズムで最適に決定できることがわかる。 しかし、この補間は学習

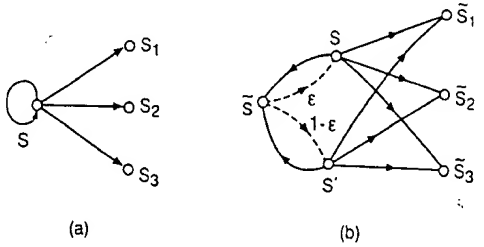



**図 6.12** 削除補間法の過程を状態図を使って表現した例

データを考慮するためというよりはむしろ、未知 (将来の) データをより良く予測するために考え出されたものであるから、$\epsilon$ の決定は、$\lambda$ と $\lambda'$ の2つのモデルの、どちらを得る場合にも使われなかったデータに基づいてなされるべきである。 よって削除補間法では、学習データ $\mathcal{T}$ を重なりのない2つの集合、つまり $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$ に分割するということが大きなポイントである。 例えば、ある場合には、$\mathcal{T}_1$ が学習データ $\mathcal{T}$ の90%、$\mathcal{T}_2$ が残りの10%になるように分けようと考えるかもしれない。 初めに、学習セット $\mathcal{T}_1$ を $\lambda$ と $\lambda'$ の学習に使う。 次に学習セット $\mathcal{T}_2$ を、$\lambda$ と $\lambda'$ が固定されていると仮定して $\epsilon$ の推定に使う。 明らかに、このような分割方法はたくさんあるが、そのなかで特に簡単な方法は $\mathcal{T}_2$ をデータ中で循環させる方法である。つまり、最初の分割ではデータの最後の10%を $\mathcal{T}_2$ として使い、次の分割ではデータの最後から1つ前の10%を $\mathcal{T}_2$ として使う、などである。 削除補間法の解釈は簡単である。 仮に、未知データがより精密なモデルによく適合したならば (よって $\lambda$ の信頼度が確認されたとき)、前向き後ろ向きアルゴリズムによって $\epsilon$ の値は1に近くなるだろう。 そうでない場合は、前向き後ろ向きアルゴリズムによって $\epsilon$ の値は小さくなる。これは、より信頼度の高いモデル $\lambda'$ の方が $\lambda$ よりも新しいデータをよく表現していることを示している。

削除補間法は、言語モデルとしての3つ組み単語確率の推定 [13]、3つ組み音素モデルの HMM の出力確率の推定など、音声認識の数多くの問題に応用され成功を収めている (この本の8章で議論する)。

### 6.13.2 ベイズ適応法

不十分なデータの問題は、話者を特定した限られた量の学習データを用いて、特定話者モデルを推定する場合にも生じる。 この問題に対するアプローチを話者適応を通して考える。話者適応は、信頼度高く学習した不特定話者モデルを、特定話者の学習データを使って、その話者に適応化する問題である [31]。



話者適応はベイズの枠組みに基づいて行なうことができる。 HMM 確率尺度 $P(\mathbf{O}|\lambda)$ を考えよう。 HMM パラメータ $\lambda$ が、固定であるが未知であると仮定するならば、$\lambda$ に対する最尤 (ML) 推定値は、学習系列 $\mathbf{O}$ が与えられたとき、次の尤度式を解くことによって得られる。

$$\frac{\partial}{\partial \lambda} P(\mathbf{O}|\lambda) = 0 \tag{6.115}$$

(通常、Baum-Welch 再推定アルゴリズムは、式 (6.115) を直接解くというよりは、ある定常的な点の解を得るために用いられる)。 もし、$\lambda$ が事前分布 $P_0(\lambda)$ に従った不規則な値であるとするならば、$\lambda$ に対する最大事後確率 (MAP) 推定値は、学習系列 $\mathbf{O}$ が与えられたとき、

$$\frac{\partial}{\partial \lambda} P(\lambda|\mathbf{O}) = 0 \tag{6.116}$$

を解くことで得られる。ベイズ理論を使って、$P(\lambda|\mathbf{O})$ は、

$$P(\lambda|\mathbf{O}) = \frac{P(\mathbf{O}|\lambda)P_0(\lambda)}{P(\mathbf{O})} \tag{6.117}$$

のように書き換えることができる。これにより、解を求める過程における事前パラメータ $P_0(\lambda)$ の影響が明らかになる。 仮に、分布が正しく選択されたならば、MAP の解は最小ベイズ危険率を満たす。

パラメータの事前分布は、注目しているパラメータの統計量を、いかなる測定もなされない前に特徴づけるものである。 パラメータ値がどのようになるか、事前分布がなんの傾向も示さないとき、その事前分布は情報のない事前分布と呼ばれる (これは基本的にパラメータ空間全体に対して一様な分布である)。 この場合、式 (6.116) を解いて得られる MAP 推定値は、式 (6.115) の ML 推定値と同じになる。 最小ベイズ危険率を満たす MAP 推定において重要なのは、モデルパラメータ値に関して事前知識がある場合に、その事前知識を事前分布の形で統合することであろう。 このタイプの事前分布はしばしば、情報のある事前分布と呼ばれる。 直感的に言うと、我々がもし、パラメータ値がどのようになるか、観測する前に知っているならば、希望的には、限られている学習データを良いモデル

推定のために、上手に利用することができるだろう。これを真とするならば、残された問題は、いかに情報のある事前分布を引きだすか、そしてそれをMAP推定値を得るためにどのように用いるかである。

ベイズ適応では、数学的に取り扱いやすくするために、共役事前分布がしばしば用いられる。不規則ベクトルの共役事前分布は、不規則ベクトルの確率密度関数のパラメータに対する事前分布として定義される。よって、事後分布 $P(\lambda|\mathbf{O})$ と事前分布 $P(\lambda)$ は、任意の観測サンプル $\mathbf{O}$ に対して同じ分布族に属する。例えば、ガウス分布の平均値の共役事前分布は、同じくガウス分布であることがよく知られている。よって、これ以降、共役事前分布を使うことだけを議論する。また、少量学習セットの問題に対処するためのベイズ適応の考え方を示すためには、ガウス平均のベイズ適応の場合だけで十分であるので、これだけについて議論する。

混合確率密度HMMのガウス混合要素 $\mathcal{N}(\mu,\sigma^2)$ に焦点を当てよう。簡単のため、1次元の観測事象を用いる。平均値 $\mu$ は事前分布 $P_0(\mu)$ に従う不規則な値で、分散値 $\sigma^2$ は既知で固定であるとする。$\mu$ に対する共役事前分布はガウス分布であると示される。つまり、もし $P_0(\mu)$ を $\mu$ の共役事前分布とするならば、$P_0(\mu)$ はガウス分布である。そこで、$\mu$ の事前分布の平均値、分散値をそれぞれ $\rho$ と $\tau^2$ で表現しよう。ベイズ適応において、$n$ 個の学習観測集合から得られる平均値パラメータ $\mu$ のMAP推定値は次のようになる。

$$\hat{\mu}_{\mathrm{MAP}} = \frac{n\tau^2}{\sigma^2+n\tau^2}\bar{o} + \frac{\sigma^2}{\sigma^2+n\tau^2}\rho \tag{6.118}$$

ここで、$\bar{o}$ は $n$ 個の学習データのサンプル平均である。式(6.118)の解釈は以下のようである。学習データが得られなかった場合は、$n=0$ なので、$\mu$ の最も良い推定値は、単に $\mu$ パラメータの事前分布の平均値 $\rho$ になる。学習データが収集され使用されるならば、MAP推定値は、事前分布の平均値 $\rho$ と、提示された観測物のサンプル平均 $\bar{o}$ の重み付き平均になる。究極的には、$n\to\infty$ となって $\mu$ の最も良い推定は、予想したようにサンプル平均 $\bar{o}$ になる。もし、事前分散 $\tau^2$ が $\sigma^2/n$ よりもずっと大きいならば、式(6.118)のMAP推定値は基本的にML推定値 $\bar{o}$ になることも注意すべきである。これは、情報のない事前分布を用いた場合に対応する。

重要なのは、どのようにして $\rho$ と $\tau^2$ を決定するかである。実際には、これらの事前パラメータは、特定話者(あるいは複数話者)モデルの集合や、各状態を混合分布で表現した不特定話者モデルから推定するのが適切である。例えば、$\rho$ や $\tau^2$ は以下のように推定できる。



$$\rho = \sum_{m=1}^{M} c_m \rho_m \tag{6.119a}$$

$$\tau^2 = \sum_{m=1}^{M} c_m (\rho_m - \rho)^2 \tag{6.119b}$$

ここで、$c_m$ は $m$ 番目のモデル(あるいは不特定話者混合分布HMMの、対応する状態の $m$ 番目の混合分布要素)に付与された重みであり、$\rho_m$ は $m$ 番目のモデル(または混合分布要素)の平均である。不特定話者ガウス混合HMMを用いる場合、重み $c_m$ は基本的に $m$ 番目の混合分布要素に対する混合重み係数であり、式(6.119)の推定値は、特定話者の学習データが観測される前の $\mu$ の平均と分散パラメータのML推定値である。

共役事前分布に基づくベイズ適応の概念は、他のパラメータに対しても同様に適用することができる。この適応法は、特定話者の学習トークンが極端に制限された場合においても、良いパラメータ推定ができる。ベイズ適応法は、特に、少数の学習トークンしか得られない場合に、直接的に学習するよりも、認識精度を大きく改善できることが実験によって示されている[30]。

### 6.13.3 誤り訂正学習

統計的パターン認識では、クラス事前分布 $P(v)$ と条件付き分布 $P(\mathbf{O}|v)$ の正確な知識によって条件付けられた最小ベイズ危険率が、認識装置の性能の理論的限界である。両方の分布が正確にわからないとき、さらに、識別器を有限個の学習セットをもとに設計する必要があるとき、誤り率を減少させようとするいくつかの方法がある。1つの方法は、判別分析と分布推定を理論的に結合した方法に基づくものである[32]。これは、学習データセットに対して識別誤り率が最小になるように識別器(判別関数)を設計する考え方である。特に、$P(v)$ や $P(\mathbf{O}|v)$ の推定値を用いて、判別関数を設計するのと同様な方法で、学習データセットに対して誤り率を最小にする識別器を設計することを考える。$P(\mathbf{O}|v)$ をモデル化するために、他の判別関数ではなくてHMM ($P(\mathbf{O}|\lambda_v)$) を使う理由は、HMMの利点(首尾一貫性、柔軟性、計算の容易さ)を利用するためである。

Bahl らは誤り訂正戦略を初めて提案した[33]。彼らは、誤り訂正学習と名付け、特に、識別誤り問題を扱った。彼らの学習アルゴリズムは、線形識別器のための誤り訂正学習法との類似性から動機づけられている。彼らの提案した手法では観測分布は、離散型 $B = [b_i(k)]$ である。ここで、$b_i(k)$ は、HMM 情報源が状態 $i$ にあるとき、ベクトル量子化符号インデックス (音響ラベル) $k$ を観測する確率である。各 $b_i(k)$ は、前向き後ろ向きアルゴリズムを用いて、符号インデックスに対する重み付き出現頻度として得ることができる。Bahl らの誤り訂正学習アルゴリズムは、以下のように動作する。まず、ラベル付けした学習データセットを用いて、前向き後ろ向きアルゴリズムから HMM パラメータ $\Lambda = \{\lambda_v\}$ を推定する。例えば、$v$ とラベル付けされている各発声 O に対し、正しいクラス $v$ に対して $P(\mathbf{O}|\lambda_v)$ を、他の誤りクラス $w$ に対して $P(\mathbf{O}|\lambda_w)$ を評価する (誤りクラスに対する尤度の評価は、すべてのクラスに対して尽くす必要はない)。$\log P(\mathbf{O}|\lambda_w) > \log P(\mathbf{O}|\lambda_v) - \Delta$ となるすべての発声に対して ($\Delta$ は定められたしきい値)、$\lambda_v$ と $\lambda_w$ を以下のように変更する。1) 正しいクラス $v$ と誤りクラス $w$ に対して、それぞれの推定値、$b'_i(k)$ と $b''_i(k)$ を得るために、ラベル付けした発声 O だけを用いて、前向き後ろ向きアルゴリズムを適用する。2) $\lambda_v$ の中の、もとの $b_i(k)$ を $b_i(k)+\gamma b'_i(k)$ に、$\lambda_w$ の中の $b_i(k)$ を $b_i(k) - \gamma b''_i(k)$ に変更する。状態が、あるモデル間で結びになっている場合、上記の処理は、もとの $b_i(k)$ を $b_i(k) + \gamma(b'_i(k) - b''_i(k))$ に置き換えるのと等価である。適応パラメータ $\gamma$ は "収束の速度" を制御し、しきい値 $\Delta$ は "ニアミス" の場合を定義する。従って、この誤り訂正学習アルゴリズムは、単語を判別するためのモデルの最も重要な部分に焦点を当てているので、明らかに最尤推定の原理とは異なる。

Bahlらは誤り訂正学習法が、最大相互情報量基準や条件付き最尤基準を用いて得られたモデルよりも (孤立単語認識タスクにおいて) 優れた性能を示すと報告している。しかし、この方法は根本的に実験的である。

片桐らは、他のいくつかの判別学習の形式を、認識誤り最小化のための、関連する学習 / 訓練の考え方を分析する枠組みと共に提案した[34]。5.6.3 節に記述した判別学習法は、容易に HMM 学習に適用することができる。Bahl らの誤り訂正アルゴリズムは、定められたリスク関数を最小にするための可能な 1 つの選択に過ぎないと言える。



## 6.14 モデルのクラスタ化と分割

統計的モデル化における 1 つの基本的な仮定は、情報源からの観測事象の変化を、仮定した統計分布でモデル化できるということである。音声認識では、情報源は 1 個の単語、音素に似たサブワードユニット、または単語系列である。生成過程の変動 (例えば、アクセント、発声速度) や処理過程の変動 (例えば、伝送歪みや雑音) から、1 つ以上の HMM を用いて情報源を表現した方が都合が良い場合がしばしばある。この複数 HMMアプローチの背景には 2 つの動機がある。第 1 に、異なる種類のデータ情報源からのすべての変動をひとまとめにすると、不必要にモデルを複雑にし、モデルの精度を低くする場合があること、第 2 に、ある種の情報源の不均一性や変動が、事前にわかっている場合があり、その場合は、データ情報源の集合と分離してモデル化した方が正当であることである。

ベクトル量子化器の設計に広く使われている $K$-平均クラスタリングアルゴリズムや、一般化ロイド (Lloyd) アルゴリズム[35]、集合の分割や決定木の設計で考えだされた貪欲 (greedy growing) アルゴリズム[36] など、いくつかの一般化クラスタリングアルゴリズムが存在する。これらのアルゴリズムは、不均一な学習データをより均一なサブグループに分割し、1 つの HMM でもより良くモデル化できるようにするのに適している。これらのクラスタリングアルゴリズムで要求される最近隣規則では、

$$P(\mathbf{O}|\lambda_i) = \max_j P(\mathbf{O}|\lambda_j) \tag{6.120}$$

のとき、観測系列 O がクラスタ $i$ に割り当てられる。ここで、$\lambda_j$ は各クラスタのモデルを示す。この簡単な最尤推定基準を使ったモデルクラスタリングアルゴリズムを音声認識に適用し、成功した例が報告されている。

モデルクラスタリングの別の方法には、与えられた音声情報源を、ある特定の特徴をもつ多数のサブクラスに任意に細分化した後、情報源の尤度を考慮してモデルを統合する一般化手法を考える方法もある。例えば、大語彙音声認識ではしばしば、認識のための特別な (文脈依存の) 単位を構築することを試みる。例えば、単位の直前 (左文脈) や直後 (右文脈) の音素の種類を関数とするような単位を構築することが考えられる。英語では、そのような単位は 10,000 のオーダーで存在する。しかし、その中の多くの単位は機能的にはほとんど同じである。問題は、(モデル数をより扱いやすくし、パラメータ推定の分散を小さくするために) どの単位どう

しを統合すべきかを決定することである。この手法を理解するために、学習観測集合 $\mathbf{O}_a$ と $\mathbf{O}_b$ に対応する2つの異なるモデル $\lambda_a$ と $\lambda_b$ を考えよう。また、統合した観測集合 $\{\mathbf{O}_a, \mathbf{O}_b\}$ に対応する統合モデル $\lambda_{a+b}$ を考えよう。モデルを統合することによるエントロピーの変化(つまり、情報の損失量)は、次のように計算することができる。

$$\begin{aligned}\Delta H_{ab} &= H_a + H_b - H_{a+b} \\ &= -P(\mathbf{O}_a|\lambda_a)\log P(\mathbf{O}_a|\lambda_a) - P(\mathbf{O}_b|\lambda_b)\log P(\mathbf{O}_b|\lambda_b) \\ &\quad + P(\{\mathbf{O}_a, \mathbf{O}_b\}|\lambda_{a+b})\ \log P(\{\mathbf{O}_a, \mathbf{O}_b\}|\lambda_{a+b}) \qquad (6.121)\end{aligned}$$

$\Delta H_{ab}$ が十分に小さいということは、モデルを統合することによるエントロピーの変化がシステムの性能に(少なくとも学習セットに対しては)悪影響を及ぼさず、かつ、モデルを統合できることを意味する。どのくらい小さければ許容できるかは、各アプリケーションに依存する。このモデル統合化手法はLeeによって、大語彙音声認識のための一般化3つ組み音素モデル集合を生成する手法として使用され成功を収めた[37]。

## 6.15 孤立単語認識のためのHMMシステム

この章で議論された手法をわかりやすくするために、HMMを用いた孤立単語認識装置を構築することを考える[38]。いま、$V$ 単語の認識語彙があり、各単語を別々のHMMでモデル化すると仮定する。さらに、記法の簡略化のため、語彙中の各単語に対し、(1人あるいは複数の話者によって) $K$ 回単語を発声した学習セットがあると仮定する。各発声は単語の(スペクトル的、または/そして、時間的)特徴を適切に表現する観測系列を構成しているとする。孤立単語音声認識を実現するために、以下の事項を実行しなければならない。

1. 語彙中の各単語 $v$ に対しHMM $(\lambda_v)$ を構築する。つまり、$v$ 番目の単語の学習観測ベクトルに対し、尤度を最適化するモデルパラメータ $(A, B, \pi)$ を推定する。

2. 認識すべき各未知単語に対し、図6.13に示す処理を実行する。つまり、単語音声の特徴分析を経て観測系列 $\mathbf{O} = \{\mathbf{o}_1\,\mathbf{o}_2 \ldots \mathbf{o}_T\}$ を得た後、すべてのモデルに対する尤度 $P(\mathbf{O}|\lambda_v)$, $(1 \le v \le V)$ を計算し、最後に、モデル尤度が最も高い単語を選択する。すなわち、



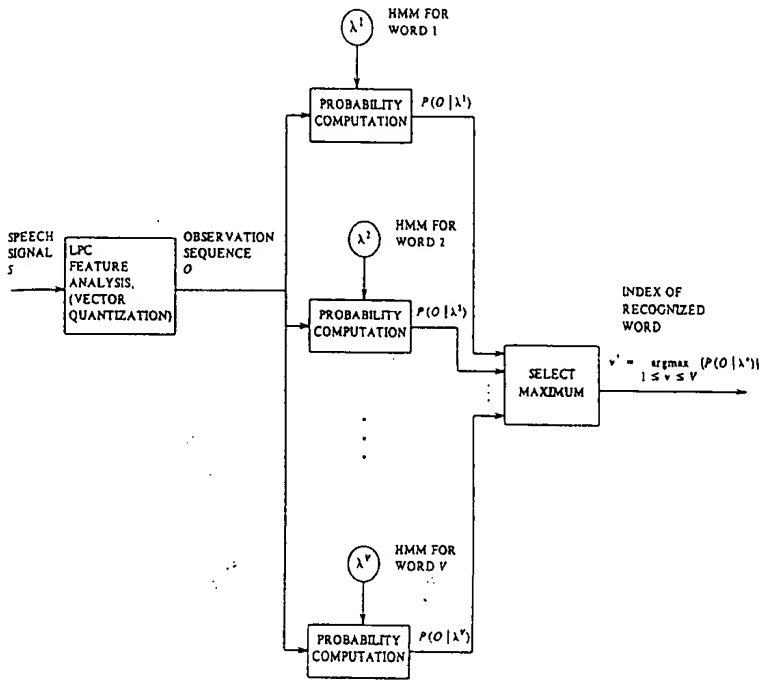



図 6.13 HMM孤立単語認識装置のブロック図(Rabinerによる[38])

$$v^* = \underset{1 \le v \le V}{\operatorname{argmax}} [P(\mathbf{O}|\lambda_v)] \qquad (6.122)$$

確率計算ステップは一般に、ビタービアルゴリズムを用いて行なわれるので(つまり、最尤パスが使われる)、$V \cdot N^2 \cdot T$ のオーダーの計算量が必要になる。小規模の語彙サイズ、例えば $V = 100$ 個の単語が、それぞれ $N = 5$ 状態の単語モデルで表現されているとき、$T = 40$ 個の観測事象を有する未知単語を認識するためには、全体で $10^5$ の計算が必要になる(各計算は積和演算、観測確率密度 $b(\mathbf{o})$ の計算である)。この計算量は、最新のほとんどの信号処理チップの能力に比べると明らかに小規模である。

### 6.15.1 モデルパラメータの選択

ここで再び、この章で何度か取り上げた問題に戻る。すなわち、モデル

のタイプをどのように選ぶか、選択したモデルのパラメータをどのように選ぶかである。 孤立単語認識において、語彙中の各単語を別々の HMM を使って設計する場合は、エルゴディックモデルよりも left-right モデルの方が適切であることは明らかであろう。なぜなら、時刻とモデルの状態とをかなり容易に関連づけられるからである (なぜなら、左から右への状態遷移が時刻と連動しているからである)。 さらに、モデルの状態の物理的な意味が、モデル化される単語中の異なる音 (例えば、音素や音節) として明確にできるからである。

各単語モデルに使用する状態数については、2 つの考え方がある。 1 つは状態数を単語中の音 (音素) の数とほぼ合わせる考え方である。この場合は、2 から 1 0 状態のモデルが適当であろう [18]。 もう 1 つの考え方は、状態数を、発声された単語中に存在する平均的な観測数にほぼ合わせる考え方で、これはいわゆる Bakis モデルである [11]。 この方法では、各状態は観測する間隔 (標準的な分析法では約 10–15ms) に関係する。 この節のあとの方で述べる結果では、前者のアプローチをとった。 さらに我々は、それぞれの単語モデルが同じ状態数になるようにした。この制約のもとでは、状態数と同じ数の音が存在する単語に対して、最もよく動作すると予想される。

単語モデルの状態数を変えた場合の効果を見るために、孤立数字認識 (すなわち、語彙 10 単語) における平均単語誤認識率を、状態数 $N$ についてプロットしたものを図 6.14 に示す。 誤認識は、$N = 6$ で極小点に達するが、6 に近い $N$ に対する誤認識率との差は小さく、$N$ に対してあまり敏感ではないように見える。

次の問題は、観測ベクトルの選択とその表現法である。 3 章で述べたように、連続モデルの観測ベクトルとしては、重み付き LPC ケプストラム係数、そして、重み付きケプストラム系列の微係数、または (自己相関 HMM に対しては) 自己相関係数が考えられる。離散シンボルモデルに対しては、離散シンボルを生成する符号帳を用いる。 また、連続モデルに対しては、1 状態当たり $M = 64 \sim 256$ 混合を用い、離散シンボルモデルに対しては、$M = 512 \sim 1024$ 符号語の符号帳を用いる。 連続モデルでは、全共分散行列を少ない混合分布数で用いるよりも、対角共分散行列をそれ以上の数の混合分布数で用いるほうが便利で、しばしば望ましいことがわかった。 この理由は簡単である。つまり、限られた学習データから、共分散行列の非対角要素を信頼度高く再推定することが困難だからで



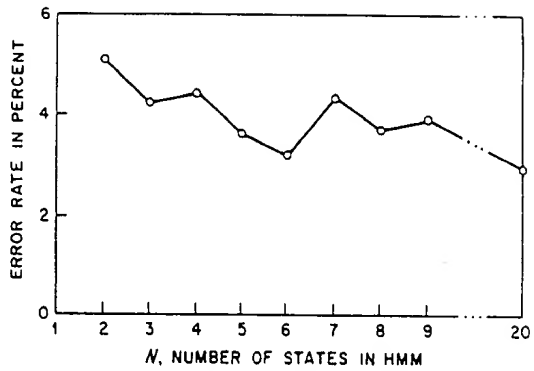



**図 6.14** (数字語彙における) HMM の状態数 $N$ と平均単語誤認識率の関係 (Rabiner らによる [18])

ある。 図 6.15 は、観測ベクトルをモデル化するためには、混合確率密度が必要であることを示している (8 次の LPC ケプストラムベクトルに、9 次目のベクトル要素として対数振幅を加えた)。 図 6.15 は、状態内の分布 $b_j(o)$ と実際の観測ベクトルのヒストグラムを比較したものである (すべての学習観測ベクトルを、最尤セグメンテーションによって、状態ごとに分割し決定した)。 観測ベクトルは 9 次元で、モデルの確率密度は $M = 5$ 混合を用いた。 各混合分布の共分散行列は、対角成分のみに制約されている。 図 6.15 は、単語 "zero" のモデルの第 1 状態に対する結果である。 第 1 パラメータ (ケプストラムの 1 次) のヒストグラムから、明らかに、$M > 1$ の値が必要なことがわかる。このパラメータは、本質的に多モードである。同様に、2 次、4 次、8 次のケプストラムにおいて、実際のデータとよく適合させるためには、混合ガウス分布が必要なことがわかる。多くのパラメータは単一ガウス分布でもよく適合しているように見えるが、いくつかの場合では、$M = 5$ でも十分に合っていない。

もう 1 つ HMM に関して実験的に確かめられた事項は、いくつかのパラメータは推定の際に、値が小さくなり過ぎないようにすることが重要であることである。 例えば、離散シンボルモデルでは、$b_j(k)$ がある最小値 $\epsilon$ に等しいか、それよりも大きくなるようにすることが必要である。これにより、学習観測集合に対し、ある状態 $j$ で $k$ 番目のシンボルが 1 度も発生しなかった場合でも、未知の観測集合をスコアリングする際に、常に、ある小さな生起確率が割り当てられることが保証される。 この点を説明する

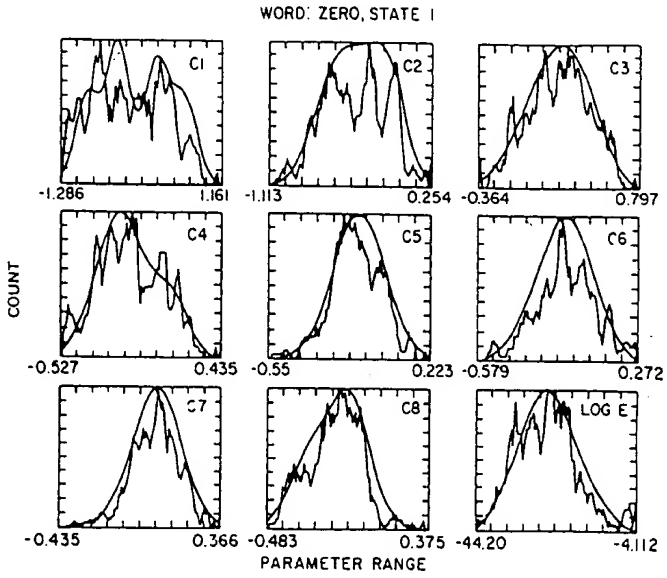



図 6.15 推定された確率密度 (ぎざぎざの輪郭) とモデルの確率密度 (滑らかな輪郭) の比較。数字 "zero" の状態 1 における観測ベクトルの 9 要素 (8 次のケプストラムと対数振幅) に対する分布を示す。(Rabiner らによる [38])

ため、標準的な単語認識実験において、パラメータ $\epsilon$ を (対数スケール軸で) 変化させたときの平均単語誤認識率の曲線を図 6.16 に示す。この図から、広範囲に渡って ($(10^{-10} \leq \epsilon \leq 10^{-3})$)、平均誤認識率がおおよそ一定になっていることがわかる。しかし、$\epsilon$ が 0 (つまり $10^{-\infty}$)) のとき、誤認識率は急激に高くなる。同様に、連続確率密度においても、混合重み $c_{jm}$ や対角共分散係数 $U_{jm}(r, r)$ が、ある最小値に等しいか、それよりも大きくなるようにすることが重要である (我々はすべての場合において $10^{-4}$ とした) [18]。

### 6.15.2 状態へのセグメンタル $K$- 平均セグメンテーション

この章で我々は、確率密度 $b_j(\mathbf{o}_t)$ の初期値を適切に推定することが、再推定式を速く、適切に収束させるために重要であることを強調してきた。この理由から、これらのパラメータの良い初期値を推定するための手法が考案された。これを図 6.17 に示す。学習手続きは、データをクラスタリングする手法としてよく知られている $K$- 平均繰り返し手法を変形したも



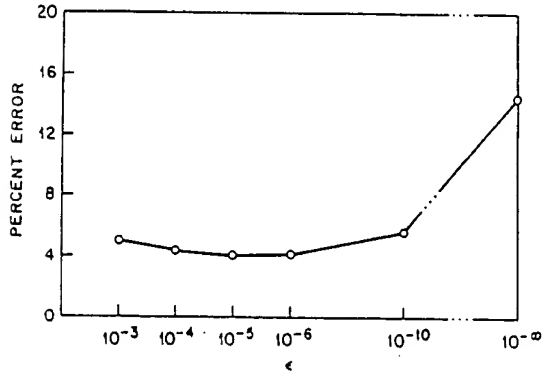



図 6.16 最小離散確率密度値 $\epsilon$ を変化させたときの平均単語誤認識率 (Rabiner らによる [18])

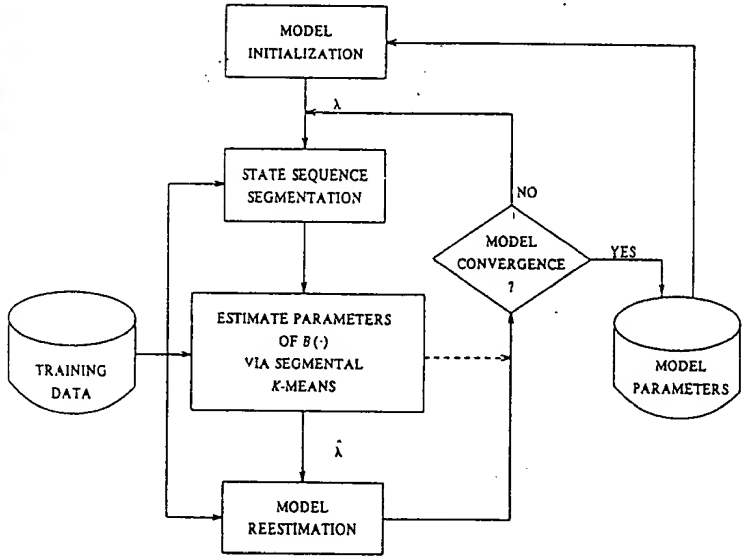



図 6.17 有限個の学習系列に適合する最適な混合連続確率密度のパラメータ値を推定するためのセグメンタル $K$- 平均学習法 (Rabiner らによる [38])

のである。

いま、(パラメータ再推定で要求されるのと同様に) 観測事象の学習セットと、すべてのモデルパラメータの初期値があるとする。 しかし、再推定で要求されるのと違い、初期モデルの推定値は不規則に選ぶか、あるいはデータに対して適切である利用可能な任意のモデルに基づいて選ばれる。

モデルの初期化に続き、学習観測系列の集合を、現在のモデル $\lambda$ に基づいて状態ごとに分割する。 この分割は、最適状態系列をビタービアルゴリズムによって決定したのち、最適パスにしたがってバックトラックして行なわれる。 この手続きを図 6.18 に示す。この図は単語 "six" の 1 回の発声における、対数振幅値のプロット、累積対数尤度のプロット、状態分割結果を示している。 図 6.18 から、各状態は発声された単語 "six" の音におおよそ対応していることがわかる。

各学習系列を分割した結果は、現在のモデルに従ったときに、$N$ 状態のそれぞれの状態 $j$ で生起すると思われる観測事象集合の最尤推定になっている。 離散シンボル確率密度を用いる場合、状態内の観測ベクトルは、$M$ 個の符号語の符号帳を用いて符号化される。よって、パラメータ $b_j(k)$ の推定値は次式によって更新される。

$$\hat{b}_j(k) = \frac{\text{状態 } j \text{ 内の符号 } k \text{ のベクトル数}}{\text{状態 } j \text{ 内のベクトル数}}$$

連続観測確率密度を使った場合、各状態 $j$ 内の観測ベクトルは、セグメンタル $K$-平均法によって、$M$ 個のクラスタに (ユークリッド歪み尺度を使って) 分割される。それぞれのクラスタは、確率密度 $b_j(\mathbf{o}_t)$ の $M$ 個の混合分布のうちの 1 つを表現する。 クラスタリング結果から、モデルパラメータ集合を以下のように更新する。

$$\hat{c}_{jm} = \frac{\text{状態 } j \text{ のクラスタ } m \text{ に分類されたベクトル数}}{\text{状態 } j \text{ 内のベクトル数}}$$

$$\hat{\boldsymbol{\mu}}_{jm} = \text{状態 } j \text{ のクラスタ } m \text{ に分類されたベクトルのサンプル平均}$$

$$\hat{\mathbf{U}}_{jm} = \text{状態 } j \text{ のクラスタ } m \text{ に分類されたベクトルのサンプル共分散行列}$$

この状態分割に基づいて、状態 $i$ から状態 $j$ へ遷移する数を数え、これを状態 $i$ からあらゆる状態 (自己ループも含む) へ遷移する数で割ることにより $a_{ij}$ の推定値を得る。

新しいモデルパラメータによって更新されたモデル $\hat{\lambda}$ が得られた後に、



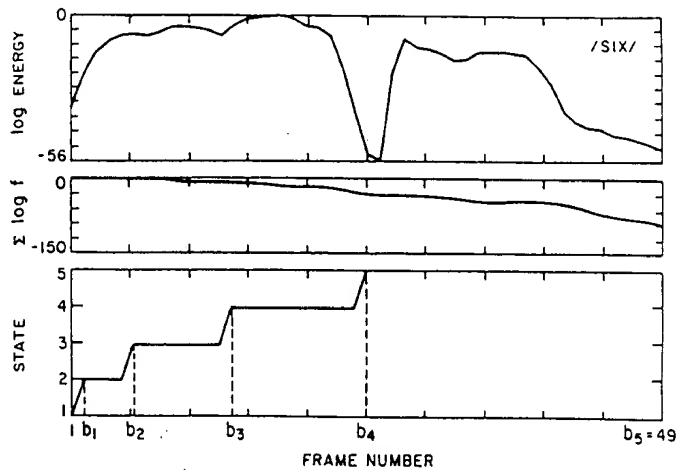



**図 6.18** 単語 "six" の発声に対する (a) 対数振幅値、(b) 累積対数尤度、および (c) 状態の割り当て結果 (Rabiner らによる [38])

すべてのモデルパラメータを再推定するために、正式な再推定手続きを実行する。 得られたモデルを更新前のモデルと (HMM の統計的類似度を反映した距離スコアを使って) 比較する。 モデルの距離スコアがしきい値を越えていたならば、古いモデル $\lambda$ を新しい (再推定された) モデル $\bar{\lambda}$ に置き換え、再び学習ループ全体を繰り返す。 モデル距離スコアがしきい値以下ならば、モデルが収束したとみなし、最後に得られたモデルパラメータを保存する。

### 6.15.3 状態継続時間長の HMM への組み込み

6.9 節では、状態継続時間長情報を HMM の技法に取り入れる理論的に正しい方法について議論した [39]。 また、継続時間長確率密度関数を HMM に含めると、コストがかなり高くなることも示した。すなわち、計算量は $D^2$ 倍に、記憶量は $D$ 倍に増加する。 $D = 25$ の値を用いると (この程度は単語認識で必要)、計算量の増加から、継続時間長制御手法がもはや導入する価値のないものになってしまう。 このことから、状態継続時間長情報を HMM に取り入れる別の手法が定式化された。これを以下に示す。

この別の手法は、前節のセグメンタル $K$-平均法を用いて学習系列を分

割した結果から、状態継続時間長確率$p_j(d)$を直接計算するものである。よって、$p_j(d)$の推定値はヒューリスティックなものである。図6.19に単語"six"の5状態モデルから得られた典型的な$p_j(d)$のヒストグラムを示す(この図は、絶対時間長$d$ではなく、正規化時間長$(d/T)$でプロットしたヒストグラムである)。最初の2つの状態は"six"の最初の/s/を表し、第3状態は母音/i/への遷移部分を表している。第4状態は母音を、第5状態は破裂音と最後の/s/の音を表していることがわかる。

ヒューリスティックな継続時間長確率密度は、認識装置において以下のように用いられた。初めに、通常のビタービアルゴリズムでバックトラックをすることにより、未知単語の観測系列を状態ごとに最適に分割する。その状態分割をもとに、各状態の継続時間長を測定する。そして、ビタービアルゴリズムの対数尤度スコアに、次のような量を事後的に加える。

$$\log \hat{P}(\mathrm{q},\ \mathrm{O}|\lambda) = \log P(\mathrm{q},\ \mathrm{O}|\lambda) + \alpha_d \sum_{j=1}^{N} \log\left[p_j(d_j)\right] \qquad (6.123)$$

ここで、$\alpha_d$は状態継続時間長スコアに対するスケーリング係数であり、

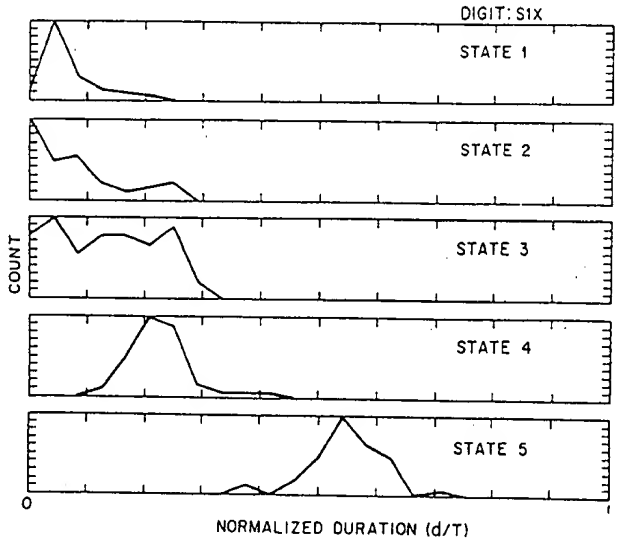



**図 6.19** 単語"six"の5状態に対する正規化継続時間長確率密度のヒストグラム(Rabiner らによる[38])



$d_j$はビタービアルゴリズムによって決定した最適パスに沿ったときの状態$j$の継続時間長である。継続時間長に対する事後処理のコスト増加分は、基本的に無視できるほど小さい。実験により、認識性能は、理論的に正しい継続時間長モデルを使った場合と基本的に同等であることがわかった。

### 6.15.4 HMM による孤立数字認識の性能

不特定話者、孤立数字認識のタスクに対する一連の認識結果を(平均単語誤認識率の観点から)示して、HMM を用いた孤立単語認識に関するこの節を結ぶ。このタスクに対する学習セットには、各数字について100人の話者が発声した100個の音声を用いた(つまり、各話者が各数字を1回発声した)。話者の半数は男性で、半数は女性である。アルゴリズムを評価するために、初期学習セット、および以下の特徴を持つ3つの独立した評価セットを用いた。

TS2: 学習に用いた話者と同じ100人; 各数字100個

TS3: 新しい話者100人のセット(男性50人、女性50人); 各数字100個

TS4: 別の新しい話者100人のセット(男性50人、女性50人); 各数字100個

認識実験の結果を表6.1に示す。認識装置は以下の通りである。

LPC/DTW: 動的計画法(DTW)を用いた従来のテンプレートベースの認識装置

**表 6.1** 種々の認識装置、評価セットに対する平均数字誤認識率

| | 評価セット | | | |
|---|---|---|---|---|
| 認識装置のタイプ | 学習セット | TS2 | TS3 | TS4 |
| LPC/DTW | 0.1 | 0.2 | 2.0 | 1.1 |
| LPC/DTW/VQ | – | 3.5 | – | – |
| HMM/VQ | – | 3.7 | – | – |
| HMM/CD | 0 | 0.2 | 1.3 | 1.8 |
| HMM/AR | 0.3 | 1.8 | 3.4 | 4.1 |

LPC/DTW/VQ: 特徴ベクトルをベクトル量子化した従来の認識装置 (M = 64)

HMM/VQ: M = 64 の符号長を用いた離散型 HMM 認識装置

HMM/CD: 1状態当たり M = 5 混合の連続確率密度モデルを用いた HMM 認識装置

HMM/AR: 自己回帰観測確率密度を用いた HMM 認識装置

表 6.1 の結果から孤立単語認識装置の性能は、VQ を用いた場合、従来手法、HMM の両モードで劣化することがわかる。 また、従来のテンプレートベースの認識装置と連続確率密度モデルの HMM 認識装置の性能は、同等であることもわかる。 最後に、自己回帰確率密度 HMM は、標準的な混合確率密度モデルよりも性能が低いことがわかる。

## 6.16 まとめ

この章では、隠れマルコフモデルの理論について、簡単な概念 (離散マルコフ連鎖) から、最も洗練されたモデル (可変継続時間長、連続確率密度モデル) まで示した。 我々の目的は、基本的な数学の物理的な説明に焦点を置くことであったので、長い証明や結果の導出は避けた。そのかわり、数式の意味を解釈することを試み、どの様にして実世界のシステムに実現されるかについて第1に重点を置いた。 また、HMM の理論を音声認識の1つの簡単な問題 (すなわち孤立単語認識) に応用した例を示した。



# 参考文献

[1] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions offinite state Markov chains," *Ann. Math. Stat.*, Vol. 37, pp. 1554-1563, 1966.

[2] L. E. Baum and J. A. Egon, "An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology," *Bull. Amer. Meteorol. Soc.*, Vol. 73, pp. 360-363, 1967.

[3] L. E. Baum and G. R. Sell, "Growth functions for transformations on manifolds," *Pac. J. Math.*, Vol. 27, No. 2, pp. 211-227, 1968.

[4] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Ann. Math. Stat.*, Vol. 41, No. 1, pp. 164-171, 1970.

[5] L. E. Baum, "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes," *Inequalities*, Vol. 3, pp. 1-8, 1972.

[6] J. K. Baker, "The dragon system — An overview," *IEEE Trans. Acoust. Speech Signal Processing*, Vol. ASSP-23, No. 1, pp. 24-29, Feb. 1975.

[7] F. Jelinek, "A fast sequential decoding algorithm using a stack," *IBM J. Res. Develop.*, Vol. 13, pp. 675-685, 1969.

[8] L. R. Bahl and F. Jelinek, "Decoding for channels with insertions, deletions, and substitutions with applications to speech recogni-

tion," *IEEE Trans. Informat. Theory*, Vol. IT-21, pp. 404-411, 1975.

[9] F. Jelinek, L. R. Bahl, and R. L. Mercer, "Design of a linguistic statistical decoder for the recognition of continuous speech," *IEEE Trans. Informat. Theory*, Vol. IT-21, pp. 250-256, 1975.

[10] F. Jelinek, "Continuous speech recognition by statistical methods," *Proc. IEEE*, Vol. 64, pp. 532-536, Apr. 1976.

[11] R. Bakis, "Continuous speech word recognition via centisecond acoustic states," in *Proc. ASA Meeting* (Washington, DC), Apr. 1976.

[12] F. Jelinek, L. R. Bahl, and R. L. Mercer, "Continuous speech recognition: Statistical methods," in *Handbook of Statistics, II*, P. R. Krishnaiad, Ed. Amsterdam, The Netherlands: North-Holland, 1982.

[13] L. R. Bahl, F. Jelinek, and R. L. Mercer, "A maximum likelihood approach to continuous speech recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-5, pp. 179-190, 1983.

[14] J. D. Ferguson, "Hidden Markov Analysis: An Introduction," in *Hidden Markov Models for Speech*, Institute for Defense Analyses, Princeton, NJ, 1980.

[15] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE Trans. Informat. Theory*, Vol. IT-13, pp. 260-269, Apr. 1967.

[16] G. D. Forney, "The Viterbi algorithm," *Proc. IEEE*, Vol. 61, pp. 268-278, Mar. 1973.

[17] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc.*, Vol. 39, No. 1 pp. 1-38, 1977.



[18] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition," *Bell System Tech. J.*, Vol. 62, No. 4, pp. 1035–1074, Apr. 1983.

[19] L. A. Liporace, "Maximum likelihood estimation for multivariate observations of Markov sources," *IEEE Trans. Informat. Theory*, Vol. IT-28, No. 5, pp. 729-734, 1982.

[20] B. H. Juang, "Maximum likelihood estimation for mixture multivariate stochastic observations of Markov chains," *AT&T Tech. J.*, Vol. 64, No. 6, pp. 1235-1249, July-Aug. 1985.

[21] B. H. Juang, S. E. Levinson, and M. M. Sondhi, "Maximum likelihood estimation for multivariate mixture observations of Markov chains," *IEEE Trans. Informat. Theory*, Vol. IT-32, No. 2, pp. 307-309, Mar. 1986.

[22] A. B. Poritz, "Linear predictive hidden Markov models and the speech signal," in *Proc. ICASSP '82* (Paris, France), pp. 1291-1294, May 1982.

[23] B. H. Juang and L. R. Rabiner, "Mixture autoregressive hidden Markov models for speech signals," *IEEE Trans. Acoust. Speech Signal Processing*, Vol. ASSP-33, No. 6, pp. 1404-1413, Dec. 1985.

[24] M J. Russell and R. K. Moore, "Explicit modeling of state occupancy in hidden Markov models for automatic speech recognition," in *Proc. ICASSP '85* (Tampa, FL), pp. 5-8, Mar. 1985.

[25] S. E. Levinson, "Continuously variable duration hidden Markov models for automatic speech recognition," *Computer, Speech and Language*, Vol. 1, No. 1, pp. 29-45, Mar. 1986.

[26] L. R. Bahl, P. F. Brown, P. V. deSouza, and L. R. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Proc. ICASSP '86* (Tokyo, Japan), pp. 49-52, Apr. 1986.

[40] D. Mansour and B.H. Juang, "A family of distortion measures based upon projection operation for robust speech recognition," *Proc. ICASSP 88*, New York, NY, April 1988; also in *IEEE Trans. Acoustics, Speech, Signal Proc.*, ASSP-37 (11): 1659–1671, November 1989.

[41] D. Mansour and B.H. Juang, "The short-time modified coherence representation and noisy speech recognition," *IEEE Trans. Acoustics, Speech, Signal Proc.* ASSP-37 (6): 795–804, June 1989.

[42] O. Ghitza, "Auditory nerve representation as a front-end for speech recognition in a noisy environment," *Computer Speech and Language*, 1 (2): 109–130, December 1986.

(Original Ref.)

Literature 5

Chapter 6

# THEORY AND IMPLEMENTATION OF HIDDEN MARKOV MODELS

## 6.1 INTRODUCTION

In Chapters 4 and 5 we presented one major pattern-recognition approach to speech recognition, namely the template method. One key idea in the template method is to derive typical sequences of speech frames for a pattern (e.g., a word) via some averaging procedure, and to rely on the use of local spectral distance measures to compare patterns. Another key idea is to use some form of dynamic programming to temporally align patterns to account for differences in speaking rates across talkers as well as across repetitions of the word by the same talker. The methodology of the template approach is well developed and provides good recognition performance for a variety of practical applications.

The template approach, however, is not based on the ideas of statistical signal modeling in a strict sense. Even though statistical techniques have been widely used in clustering to create reference patterns, the template approach is best classified as a simplified, nonparametric method in which a multiplicity of reference tokens (sequences) are used to characterize the variation among different utterances. As such, statistical signal characterization inherent in the template representation is only implicit and often inadequate. Consider, for example, the use of a truncated cepstral distortion measure as the local distance for template matching. The Euclidean distance form of the cepstral distance measure suggests that the reference vector can be viewed as the *mean* of some assumed distribution.

Obviously, this simple form of the sufficient statistic[1] (use of only the mean reference vector) neglects the second-order statistics—i.e., *covariances*, which, as will be seen later, are of particular significance in statistical modeling. (Note that this distribution is used to account for variations of the cepstral coefficients at the frame level since time alignment is performed so as to match appropriate frames of the patterns being compared.) There is clearly a need to use a more elaborate and analytical statistical method for speech recognition.

In this chapter we will study one well-known and widely used statistical method of characterizing the spectral properties of the frames of a pattern, namely the hidden Markov model (HMM) approach. (These models are also referred to as Markov sources or probabilistic functions of Markov chains in the communications literature.) The underlying assumption of the HMM (or any other type of statistical model) is that the speech signal can be well characterized as a parametric random process, and that the parameters of the stochastic process can be determined (estimated) in a precise, well-defined manner. We will show that the HMM method provides a natural and highly reliable way of recognizing speech for a wide range of applications and integrates well into systems incorporating both task syntax and semantics.

The basic theory of hidden Markov models was published in a series of classic papers by Baum and his colleagues ([1]–[5]) in the late 1960s and early 1970s and was implemented for speech-processing applications by Baker [6] at CMU, and by Jelinek and his colleagues at IBM ([7]–[13]) in the 1970s.

We begin this chapter with a review of the theory of Markov chains and then extend the ideas to HMMs using several simple examples. Based on the now-classical approach of Jack Ferguson of IDA (Institute for Defense Analyses), as introduced in lectures and in writing [14], we will focus our attention on the three fundamental problems for HMM design, namely: the evaluation of the probability (or likelihood) of a sequence of observations given a specific HMM; the determination of a best sequence of model states; and the adjustment of model parameters so as to best account for the observed signal. We will show that once these three fundamental problems are solved, we can readily apply HMMs to selected problems in speech recognition.

## 6.2 DISCRETE-TIME MARKOV PROCESSES

Consider a system that may be described at any time as being in one of a set of $N$ distinct states indexed by $\{1, 2, \ldots, N\}$ as illustrated in Figure 6.1 (where $N = 5$ for simplicity). At regularly spaced, discrete times, the system undergoes a change of state (possibly back to the same state) according to a set of probabilities associated with the state. We denote the time instants associated with state changes as $t = 1, 2, \ldots$, and we denote the actual

[1]Sufficient statistics are a set of measurements from a process which contain all the relevant information for estimating the parameters of that process.



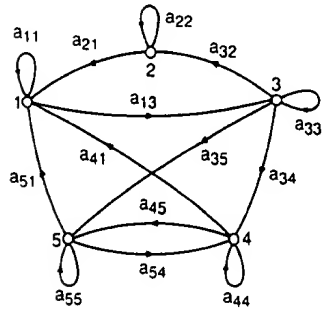



Figure 6.1 A Markov chain with five states (labeled 1 to 5) with selected state transitions.

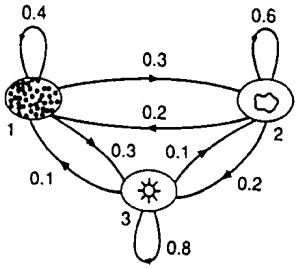



Figure 6.2 Markov model of the weather.

state at time $t$ as $q_t$. A full probabilistic description of the above system would, in general, require specification of the current state (at time $t$), as well as all the predecessor states. For the special case of a discrete-time, first order, Markov chain, the probabilistic dependence is truncated to just the preceding state—that is,

$$P[q_t = j | q_{t-1} = i,\ q_{t-2} = k, \ldots] = P[q_t = j | q_{t-1} = i]. \tag{6.1}$$

Furthermore, we consider only those processes in which the right-hand side of (6.1) is independent of time, thereby leading to the set of state-transition probabilities $a_{ij}$ of the form

$$a_{ij} = P[q_t = j | q_{t-1} = i], \qquad 1 \le i, j \le N \tag{6.2}$$

with the following properties

$$a_{ij} \ge 0 \qquad \forall j, i \tag{6.3a}$$

$$\sum_{j=1}^{N} a_{ij} = 1 \qquad \forall i \tag{6.3b}$$

since they obey standard stochastic constraints.

The above stochastic process could be called an observable Markov model because the output of the process is the set of states at each instant of time, where each state corresponds to an observable event. To set ideas, consider a simple three-state Markov model of the weather as shown in Figure 6.2. We assume that once a day (e.g., at noon), the weather is observed as being one of the following:

**State 1:** precipitation (rain or snow)

**State 2:** cloudy

**State 3:** sunny.

We postulate that the weather on day $t$ is characterized by a single one of the three states above, and that the matrix $A$ of state-transition probabilities is

$$A = \{a_{ij}\} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}.$$

Given the model of Figure 6.2 we can now ask (and answer) several interesting questions about weather patterns over time. For example, we can pose the following simple problem:

**Problem**

What is the probability (according to the model) that the weather for eight consecutive days is "sun-sun-sun-rain-rain-sun-cloudy-sun"?

**Solution**

We define the observation sequence, O, as

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| O | = | ( sunny, | sunny, | sunny, | rain, | rain, | sunny, | cloudy, | sunny ) |
| | = | ( 3, | 3, | 3, | 1, | 1, | 3, | 2, | 3 ) |
| day | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

corresponding to the postulated set of weather conditions over the eight-day period and we want to calculate $P(\mathbf{O}|\text{Model})$, the probability of the observation sequence O, given the model of Figure 6.2. We can directly determine $P(\mathbf{O}|\text{Model})$ as:

$$\begin{aligned} P(\mathbf{O}|\text{Model}) &= P[3, 3, 3, 1, 1, 3, 2, 3|\text{Model}] \\ &= P[3]\,P[3|3]^2 P[1|3]\,P[1|1] \\ &\quad P[3|1]\,P[2|3]\,P[3|2] \\ &= \pi_3 \cdot (a_{33})^2 a_{31} a_{11} a_{13} a_{32} a_{23} \\ &= (1.0)(0.8)^2(0.1)(0.4)(0.3)(0.1)(0.2) \\ &= 1.536 \times 10^{-4} \end{aligned}$$

where we use the notation:

$$\pi_i = P[q_1 = i], \qquad 1 \le i \le N \tag{6.4}$$

to denote the initial state probabilities.

Another interesting question we can ask (and answer using the model) is:

**Problem**

Given that the system is in a known state, what is the probability that it stays in that state for exactly

**Solution**

This probability can be evaluated as the probability of the observation sequence

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| O | = ( $i$, | $i$, | $i$ | , ..., | $i$, | $j \neq i$ ) | |
| day | | 1 | 2 | 3 | $d$ | $d+1$ | |



given the model, which is

$$\begin{aligned} P(\mathbf{O}|\text{Model}, q_1 = i) &= P(\mathbf{O}, q_1 = i|\text{Model})/P(q_1 = i) \\ &= \pi_i (a_{ii})^{d-1}(1 - a_{ii})/\pi_i \\ &= (a_{ii})^{d-1}(1 - a_{ii}) \\ &= p_i(d) \end{aligned} \tag{6.5}$$

The quantity $p_i(d)$ is the probability distribution function of duration $d$ in state $i$. This exponential distribution is characteristic of the state duration in a Markov chain. Based on $p_i(d)$, we can readily calculate the expected number of observations (duration) in a state, conditioned on starting in that state as

$$\bar{d}_i = \sum_{d=1}^{\infty} d p_i(d) \tag{6.6a}$$

$$= \sum_{d=1}^{\infty} d(a_{ii})^{d-1}(1 - a_{ii}) = \frac{1}{1 - a_{ii}}. \tag{6.6b}$$

Thus the expected number of consecutive days of sunny weather, according to the model, is $1/(0.2) = 5$; for cloudy it is 2.5; for rain it is 1.67.

**Problem**

Derive the expression for the mean of $p_i(d)$, i.e. Eq. (6.6b).

**Solution**

$$\begin{aligned} \bar{d}_i &= \sum_{d=1}^{\infty} d p_i(d) \\ &= \sum_{d=1}^{\infty} d(a_{ii})^{d-1}(1 - a_{ii}) \\ &= (1 - a_{ii}) \frac{\partial}{\partial a_{ii}} \left[ \sum_{d=1}^{\infty} a_{ii}^d \right] \\ &= (1 - a_{ii}) \frac{\partial}{\partial a_{ii}} \left( \frac{a_{ii}}{1 - a_{ii}} \right) \\ &= \frac{1}{1 - a_{ii}}. \end{aligned}$$

## 6.3 EXTENSIONS TO HIDDEN MARKOV MODELS

So far we have considered Markov models in which each state corresponded to a deterministically observable event. Thus, the output of such sources in any given state is not random. This model is too restrictive to be applicable to many problems of interest. In this section we extend the concept of Markov models to include the case in which the observation is a probabilistic function of the state—that is, the resulting model (which is

called a hidden Markov model) is a doubly embedded stochastic process with an underlying stochastic process that is *not* directly observable (it is hidden) but can be observed only through another set of stochastic processes that produce the sequence of observations.

To illustrate the basic concepts of the hidden Markov model, we will use several simple examples including simple coin-tossing experiments. We begin with a review of some basic ideas of probability in the following exercise.

**Exercise 6.1**

Given a single fair coin, i.e., $P(\text{Heads}) = P(\text{Tails}) = 0.5$, which you toss once and observe Tails,

1. What is the probability that the next 10 tosses will provide the sequence (*HHTHTTHTTH*)?
2. What is the probability that the next 10 tosses will produce the sequence (*HHHHHHHHHH*)?
3. What is the probability that 5 of the next 10 tosses will be tails? What is the expected number of tails over the next 10 tosses?

**Solution 6.1**

1. For a fair coin, with independent coin tosses, the probability of any specific observation sequence of length 10 (10 tosses) is $(1/2)^{10}$ since there are $2^{10}$ such sequences and all are equally probable. Thus:

$$P(HHTHTTHTTH) = \left(\frac{1}{2}\right)^{10}.$$

2.

$$P(HHHHHHHHHH) = \left(\frac{1}{2}\right)^{10}.$$

Thus a specified run of length 10 is as likely as a specified run of interlaced $H$ and $T$.

3. The probability of 5 tails in the next 10 tosses is just the number of observation sequences with 5 tails and 5 heads (in any order) and this is

$$P(5H, 5T) = \binom{10}{5}\left(\frac{1}{2}\right)^{10} = \frac{252}{1024} \cong 0.25$$

since there are $\binom{10}{5}$ ways of getting $5H$ and $5T$ in 10 tosses, and each sequence has probability of $\left(\frac{1}{2}\right)^{10}$. The expected number of tails in 10 tosses is

$$E(T \text{ in 10 tosses}) = \sum_{d=0}^{10} d\binom{10}{d}\left(\frac{1}{2}\right)^{10} = 5.$$

Thus, on average, there will be $5H$ and $5T$ in 10 tosses, but the probability of exactly $5H$ and $5T$ is only 0.25.

### 6.3.1 Coin-Toss Models

Assume the following scenario. You are in a room with a barrier (e.g., a curtain) through



which you cannot see what is happening. On the other side of the barrier is another person who is performing a coin-tossing experiment (using one or more coins). The person will not tell you which coin he selects at any time; he will only tell you the result of each coin flip. Thus a sequence of *hidden* coin-tossing experiments is performed, with the observation sequence consisting of a series of heads and tails. A typical observation sequence would be

$$\begin{aligned} O &= (o_1\, o_2\, o_3 \ldots o_T) \\ &= (H\,H\,T\,T\,T\,H\,T\,T\,H \ldots H) \end{aligned}$$

where $H$ stands for heads and $T$ stands for tails.

Given the above scenario, the question is, How do we build an HMM to explain (model) the observed sequence of heads and tails? The first problem we face is deciding what the states in the model correspond to, and then deciding how many states should be in the model. One possible choice would be to assume that only a single biased coin was being tossed. In this case, we could model the situation with a two-state model in which each state corresponds to the outcome of the previous toss (i.e., heads or tails). This model is depicted in Figure 6.3a. In this case, the Markov model is observable, and the only issue for complete specification of the model would be to decide on the best value for the single parameter of the model (i.e., the probability of, say, heads). Interestingly, an equivalent HMM to that of Figure 6.3a would be a degenerate one-state model in which the state corresponds to the single biased coin, and the unknown parameter is the bias of the coin.

A second HMM for explaining the observed sequence of coin toss outcomes is given in Figure 6.3b. In this case there are two states in the model, and each state corresponds to a different, biased coin being tossed. Each state is characterized by a probability distribution of heads and tails, and transitions between states are characterized by a state-transition matrix. The physical mechanism that accounts for how state transitions are selected could itself be a set of independent coin tosses or some other probabilistic event.

A third form of HMM for explaining the observed sequence of coin toss outcomes is given in Figure 6.3c. This model corresponds to using three biased coins, and choosing from among the three, based on some probabilistic event.

Given the choice among the three models shown in Figure 6.3 for explaining the observed sequence of heads and tails, a natural question would be which model best matches the actual observations. It should be clear that the simple one-coin model of Figure 6.3a has only one unknown parameter; the two-coin model of Figure 6.3b has four unknown parameters; and the three-coin model of Figure 6.3c has nine unknown parameters. Thus, with the greater degrees of freedom, the larger HMMs would seem to be inherently more capable of modeling a series of coin-tossing experiments than would equivalently smaller models. Although this is theoretically true, we will see later in this chapter that practical considerations impose some strong limitations on the size of models that we can consider. A fundamental question here is whether the observed head-tail sequence is long and rich enough to be able to specify a complex model. Also, it might just be the case that only a single coin is being tossed. Then using the three-coin model of Figure 6.3c would be inappropriate because we would be using an underspecified system.

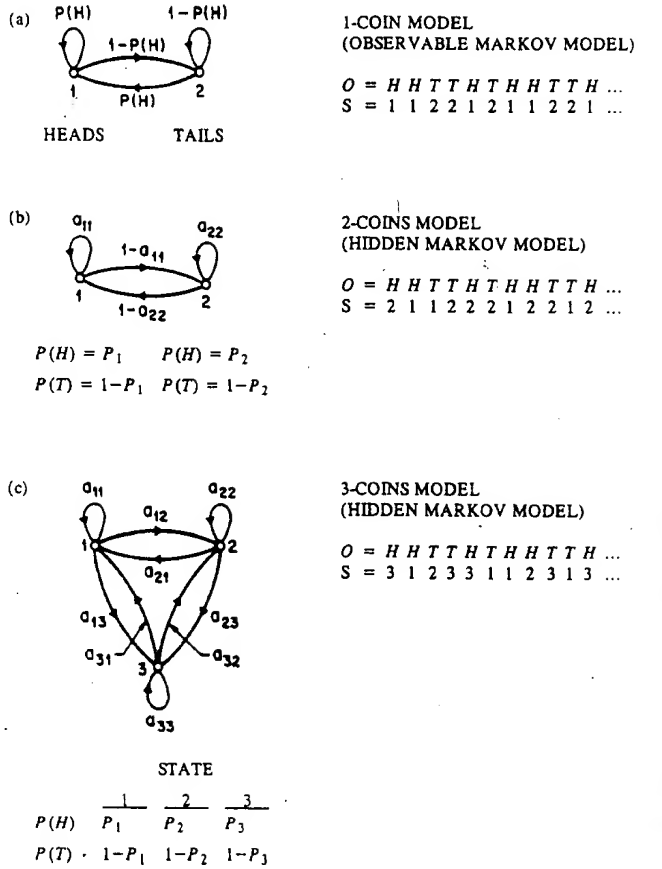



**Figure 6.3** Three possible Markov models that can account for the results of hidden coin-tossing experiments. (a) one-coin model, (b) two-coins model, (c) three-coins model.

### 6.3.2 The Urn-and-Ball Model

To extend the ideas of the HMM to a somewhat more complicated situation, consider the urn-and-ball system of Figure 6.4. We assume that there are $N$ (large) glass urns in a room. Within each urn is a large quantity of colored balls. We assume there are $M$ distinct colors of the balls. The physical process for obtaining observations is as follows. A genie is in the room, and, according to some random procedure, it chooses an initial urn. From this urn, a ball is chosen at random, and its color is recorded as the observation. The ball is



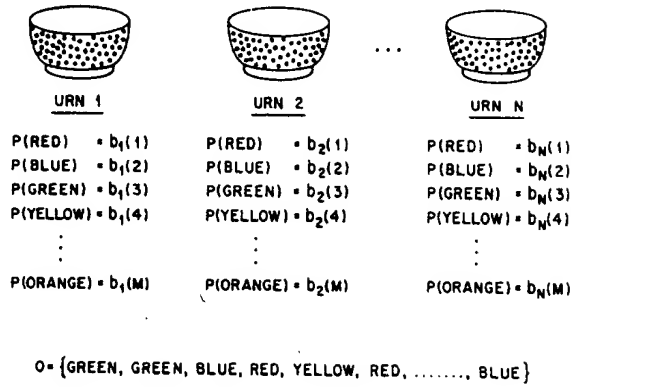



**Figure 6.4** An $N$-state urn-and-ball model illustrating the general case of a discrete symbol HMM.

then replaced in the urn from which it was selected. A new urn is then selected according to the random selection procedure associated with the current urn, and the ball selection process is repeated. This entire process generates a finite observation sequence of colors, which we would like to model as the observable output of an HMM.

It should be obvious that the simplest HMM that corresponds to the urn-and-ball process is one in which each state corresponds to a specific urn, and for which a (ball) color probability is defined for each state. The choice of urns is dictated by the state-transition matrix of the HMM.

It should be noted that the ball colors in each urn may be the same, and the distinction among various urns is in the way the collection of colored balls is composed. Therefore, an isolated observation of a particular color ball does not immediately tell which urn it is drawn from.

### 6.3.3 Elements of an HMM

The above examples give us some idea of what an HMM is and how it can be applied to some simple scenarios. We now formally define the elements of an HMM.

An HMM for discrete symbol observations such as the above urn-and-ball model is characterized by the following:

1. $N$, the number of states in the model. Although the states are hidden, for many practical applications there is often some physical significance attached to the states or to sets of states of the model. Thus, in the coin-tossing experiments, each state corresponded to a distinct biased coin. In the urn-and-ball model, the states corresponded to the urns. Generally the states are interconnected in such a way that any state can be reached from any other state (i.e., an ergodic model); however, we will see later in this chapter that other possible interconnections of states are often

of interest and may better suit speech applications. We label the individual states as $\{1, 2, \ldots, N\}$, and denote the state at time $t$ as $q_t$.

2. $M$, the number of distinct observation symbols per state—i.e., the discrete alphabet size. The observation symbols correspond to the physical output of the system being modeled. For the coin-toss experiments the observation symbols were simply heads or tails; for the ball-and-urn model they were the colors of the balls selected from the urns. We denote the individual symbols as $V = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_M\}$.
3. The state-transition probability distribution $A = \{a_{ij}\}$ where

$$a_{ij} = P[q_{t+1} = j | q_t = i], \qquad 1 \leq i, j \leq N. \tag{6.7}$$

For the special case in which any state can reach any other state in a single step, we have $a_{ij} > 0$ for all $i, j$. For other types of HMMs, we would have $a_{ij} = 0$ for one or more $(i, j)$ pairs.

4. The observation symbol probability distribution, $B = \{b_j(k)\}$, in which

$$b_j(k) = P[\mathbf{o}_t = \mathbf{v}_k | q_t = j], \qquad 1 \leq k \leq M, \tag{6.8}$$

defines the symbol distribution in state $j, j = 1, 2, \ldots, N$.

5. The initial state distribution $\pi = \{\pi_i\}$ in which

$$\pi_i = P[q_1 = i], \qquad 1 \leq i \leq N. \tag{6.9}$$

It can be seen from the above discussion that a complete specification of an HMM requires specification of two model parameters, $N$ and $M$, specification of observation symbols, and the specification of the three sets of probability measures $A, B$, and $\pi$. For convenience, we use the compact notation

$$\lambda = (A, B, \pi) \tag{6.10}$$

to indicate the complete parameter set of the model. This parameter set, of course, defines a probability measure for O, i.e. $P(\mathrm{O}|\lambda)$, which we discuss in the next section. We use the terminology HMM to indicate the parameter set $\lambda$ and the associated probability measure interchangeably without ambiguity.

### 6.3.4 HMM Generator of Observations

Given appropriate values of $N, M, A, B$, and $\pi$, the HMM can be used as a generator to give an observation sequence

$$\mathrm{O} = (\mathbf{o}_1 \mathbf{o}_2 \ldots \mathbf{o}_T) \tag{6.11}$$

(in which each observation $\mathbf{o}_t$ is one of the symbols from $V$, and $T$ is the number of observations in the sequence) as follows:

1. Choose an initial state $q_1 = i$ according to the initial state distribution $\pi$.
2. Set $t = 1$.
3. Choose $\mathbf{o}_t = \mathbf{v}_k$ according to the symbol probability distribution in state $i$, i.e., $b_j(k)$.



4. Transit to a new state $q_{t+1} = j$ according to the state-transition probability distribution for state $i$, i.e., $a_{ij}$.
5. Set $t = t + 1$; return to step 3 if $t < T$; otherwise, terminate the procedure.

The following table shows the sequence of states and observations generated by the above procedure:

| time, $t$ | 1 | 2 | 3 | 4 | 5 | 6 | ... | $T$ |
|---|---|---|---|---|---|---|---|---|
| state | $q_1$ | $q_2$ | $q_3$ | $q_4$ | $q_5$ | $q_6$ | ... | $q_T$ |
| observation | $\mathbf{o}_1$ | $\mathbf{o}_2$ | $\mathbf{o}_3$ | $\mathbf{o}_4$ | $\mathbf{o}_5$ | $\mathbf{o}_6$ | ... | $\mathbf{o}_T$ |

The above procedure can be used as both a generator of observations and as a model to simulate how a given observation sequence was generated by an appropriate HMM.

**Exercise 6.2**

Consider an HMM representation (parametrized by $\lambda$) of a coin-tossing experiment. Assume a three-state model (corresponding to three different coins) with probabilities

| | State 1 | State 2 | State 3 |
|---|---|---|---|
| $P(H)$ | 0.5 | 0.75 | 0.25 |
| $P(T)$ | 0.5 | 0.25 | 0.75 |

and with all state-transition probabilities equal to 1/3. (Assume initial state probabilities of 1/3.)

1. You observe the sequence

$$\mathrm{O} = (HHHHTHTTTT).$$

What state sequence is most likely? What is the probability of the observation sequence and this most likely state sequence?

2. What is the probability that the observation sequence came entirely from state 1?
3. Consider the observation sequence

$$\mathrm{O} = (HTTHTHHTTH).$$

How would your answers to parts a and b change?

4. If the state-transition probabilities were

$$\begin{array}{lll} a_{11} = 0.9 & , \quad a_{21} = 0.45 & , \quad a_{31} = 0.45 \\ a_{12} = 0.05 & , \quad a_{22} = 0.1 & , \quad a_{32} = 0.45 \\ a_{13} = 0.05 & , \quad a_{23} = 0.45 & , \quad a_{33} = 0.1 \end{array}$$

that is, a new model $\lambda'$, how would your answers to parts 1–3 change? What does this suggest about the type of sequences generated by the models?

**Solution 6.2**

1. Given $\mathrm{O} = (HHHHTHTTTT)$ and that all state transitions are equiprobable, the most likely state sequence is the one for which the probability of each individual observation

is maximum. Thus for each $H$, the most likely state is 2 and for each $T$ the most likely state is 3. Thus the most likely state sequence is

$$\mathbf{q} = (2222323333).$$

The probability of O and q (given the model) is

$$P(\mathbf{O}, \mathbf{q}|\lambda) = (0.75)^{10}\left(\frac{1}{3}\right)^{10}.$$

2. The probability of O given that $\hat{\mathbf{q}}$ is

$$\hat{\mathbf{q}} = (1111111111)$$

is

$$P(\mathbf{O}, \hat{\mathbf{q}}|\lambda) = (0.50)^{10}\left(\frac{1}{3}\right)^{10}.$$

The ratio of $P(\mathbf{O}, \mathbf{q}|\lambda)$ to $P(\mathbf{O}, \hat{\mathbf{q}}|\lambda)$ is:

$$R = \frac{P(\mathbf{O}, \mathbf{q}|\lambda)}{P(\mathbf{O}, \hat{\mathbf{q}}|\lambda)} = \left(\frac{3}{2}\right)^{10} = 57.67$$

which shows, as expected, that q is more likely than $\hat{\mathbf{q}}$.

3. Given $\tilde{\mathbf{O}}$ which has the same number of $H$s and $T$s, the answers to parts 1 and 2 would remain the same, as the most likely states occur the same number of times in both cases.
4. The new probability of O and q becomes

$$P(\mathbf{O}, \mathbf{q}|\lambda') = (0.75)^{10}\left(\frac{1}{3}\right)(0.1)^6(0.45)^3.$$

The new probability of O and $\hat{\mathbf{q}}$ becomes

$$P(\mathbf{O}, \hat{\mathbf{q}}|\lambda') = (0.50)^{10}\left(\frac{1}{3}\right)(0.9)^9.$$

The ratio is

$$R = \left(\frac{3}{2}\right)^{10}\left(\frac{1}{9}\right)^6\left(\frac{1}{2}\right)^3 = 1.36 \times 10^{-5}.$$

In other words, because of the nonuniform transition probabilities, $\hat{\mathbf{q}}$ is more likely than q. (The reader is encouraged to find the most likely state sequence in this case.) Now, the probability of $\tilde{\mathbf{O}}$ and q is not the same as the probability of O and q. We have

$$P(\tilde{\mathbf{O}}, \mathbf{q}|\lambda') = \frac{1}{3}(0.1)^6(0.45)^3(0.25)^4(0.75)^6$$

$$P(\tilde{\mathbf{O}}, \hat{\mathbf{q}}|\lambda') = (0.50)^{10}\left(\frac{1}{3}\right)(0.9)^9$$

with ratio

$$R = \left(\frac{1}{9}\right)^6\left(\frac{1}{2}\right)^3\left(\frac{1}{2}\right)^4\left(\frac{3}{2}\right)^6 = 1.67 \times 10^{-7}.$$

Clearly, because $a_{11} = 0.9$, $\hat{\mathbf{q}}$ is more likely.



## 6.4 THE THREE BASIC PROBLEMS FOR HMMS

Given the form of HMM of the previous section, three basic problems of interest must be solved for the model to be useful in real-world applications. These problems are the following:

**Problem 1**

Given the observation sequence $\mathbf{O} = (\mathbf{o}_1\,\mathbf{o}\ldots\mathbf{o}_T)$, and a model $\lambda = (A, B, \pi)$, how do we efficiently compute $P(\mathbf{O}|\lambda)$, the probability of the observation sequence, given the model?

**Problem 2**

Given the observation sequence $\mathbf{O} = (\mathbf{o}_1\,\mathbf{o}\ldots\mathbf{o}_T)$, and the model $\lambda$, how do we choose a corresponding state sequence $\mathbf{q} = (q_1 q_2 \ldots q_T)$ that is optimal in some sense (i.e., best "explains" the observations)?

**Problem 3**

How do we adjust the model parameters $\lambda = (A, B, \pi)$ to maximize $P(\mathbf{O}|\lambda)$?

Problem 1 is the evaluation problem; namely, given a model and a sequence of observations, how do we compute the probability that the observed sequence was produced by the model? We can also view the problem as one of scoring how well a given model matches a given observation sequence. The latter viewpoint is extremely useful. For example, if we consider the case in which we are trying to choose among several competing models, the solution to Problem 1 allows us to choose the model that best matches the observations.

Problem 2 is the one in which we attempt to uncover the hidden part of the model—that is, to find the "correct" state sequence. It should be clear that for all but the case of degenerate models, there is no "correct" state sequence to be found. Hence for practical situations, we usually use an optimality criterion to solve this problem as best as possible. As we will see, several reasonable optimality criteria can be imposed, and hence the choice of criterion is a strong function of the intended use for the uncovered state sequence. Typical uses might be to learn about the structure of the model, to find optimal state sequences for continuous speech recognition, or to get average statistics of individual states, etc.

Problem 3 is the one in which we attempt to optimize the model parameters to best describe how a given observation sequence comes about. The observation sequence used to adjust the model parameters is called a training sequence because it is used to "train" the HMM. The training problem is the crucial one for most applications of HMMs, because it allows us to optimally adapt model parameters to observed training data—i.e., to create best models for real phenomena.

To fix ideas, consider the following simple isolated-word speech recognizer. For each word of a $W$ word vocabulary, we want to design a separate $N$-state HMM. We represent the speech signal of a given word as a time sequence of coded spectral vectors. We assume that the coding is done using a spectral codebook with $M$ unique spectral vectors; hence each observation is the index of the spectral vector closest (in some spectral distortion sense) to the original speech signal. Thus, for each vocabulary word, we have a training sequence

consisting of a number of repetitions of sequences of codebook indices of the word (by one or more talkers). The first task is to build individual word models. This task is done by using the solution to Problem 3 to optimally estimate model parameters for each word model. To develop an understanding of the physical meaning of the model states, we use the solution to Problem 2 to segment each of the word training sequences into states, and then study the properties of the spectral vectors that lead to the observations occurring in each state. The goal here is to make refinements of the model (e.g., more states, different codebook size) to improve its capability of modeling the spoken word sequences. Finally, once the set of $W$ HMMs has been designed and optimized, recognition of an unknown word is performed using the solution to Problem 1 to score each word model based upon the given test observation sequence, and select the word whose model score is highest (i.e., the highest likelihood).

In the next sections we present formal mathematical solutions to each fundamental problem for HMMs. We shall see that the three problems are tightly linked together under the probabilistic framework.

### 6.4.1 Solution to Problem 1—Probability Evaluation

We wish to calculate the probability of the observation sequence, $\mathbf{O} = (\mathbf{o}_1\, \mathbf{o} \ldots \mathbf{o}_T)$, given the model $\lambda$, i.e., $P(\mathbf{O}|\lambda)$. The most straightforward way of doing this is through enumerating every possible state sequence of length $T$ (the number of observations). There are $N^T$ such state sequences. Consider one such fixed-state sequence

$$\mathbf{q} = (q_1 q_2 \ldots q_T) \tag{6.12}$$

where $q_1$ is the initial state. The probability of the observation sequence O given the state sequence of Eq. (6.12) is

$$P(\mathbf{O}|\mathbf{q}, \lambda) = \prod_{t=1}^{T} P(\mathbf{o}_t|q_t, \lambda) \tag{6.13a}$$

where we have assumed statistical independence of observations. Thus we get

$$P(\mathbf{O}|\mathbf{q}, \lambda) = b_{q_1}(\mathbf{o}_1) \cdot b_{q_2}(\mathbf{o}_2) \ldots b_{q_T}(\mathbf{o}_T). \tag{6.13b}$$

The probability of such a state sequence q can be written as

$$P(\mathbf{q}|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \cdots a_{q_{T-1} q_T}. \tag{6.14}$$

The joint probability of O and q, i.e., the probability that O and q occur simultaneously, is simply the product of the above two terms, i.e.,

$$P(\mathbf{O}, \mathbf{q}|\lambda) = P(\mathbf{O}|\mathbf{q}, \lambda) P(\mathbf{q}|\lambda). \tag{6.15}$$

The probability of O (given the model) is obtained by summing this joint probability over all possible state sequences q, giving

$$P(\mathbf{O}|\lambda) = \sum_{\text{all } \mathbf{q}} P(\mathbf{O}|\mathbf{q}, \lambda) P(\mathbf{q}|\lambda) \tag{6.16}$$



$$= \sum_{q_1, q_2, \ldots, q_T} \pi_{q_1} b_{q_1}(\mathbf{o}_1) a_{q_1 q_2} b_{q_2}(\mathbf{o}_2) \ldots a_{q_{T-1} q_T} b_{q_T}(\mathbf{o}_T). \tag{6.17}$$

The interpretation of the computation in the above equation is the following. Initially (at time $t = 1$) we are in state $q_1$ with probability $\pi_{q_1}$, and generate the symbol $\mathbf{o}_1$ (in this state) with probability $b_{q_1}(\mathbf{o}_1)$. The clock changes from time $t$ to $t + 1$ (time = 2) and we make a transition to state $q_2$ from state $q_1$ with probability $a_{q_1 q_2}$, and generate symbol $\mathbf{o}_2$ with probability $b_{q_2}(\mathbf{o}_2)$. This process continues in this manner until we make the last transition (at time $T$) from state $q_{T-1}$ to state $q_T$ with probability $a_{q_{T-1} q_T}$ and generate symbol $\mathbf{o}_T$ with probability $b_{q_T}(\mathbf{o}_T)$.

A little thought should convince the reader that the calculation of $P(\mathbf{O}|\lambda)$, according to its direct definition (Eq. (6.17)) involves on the order of $2T \cdot N^T$ calculations, since at every $t = 1, 2, \ldots, T$, there are $N$ possible states that can be reached (i.e., there are $N^T$ possible state sequences), and for each such state sequence about $2T$ calculations are required for each term in the sum of Eq. (6.17). (To be precise, we need $(2T - 1)N^T$ multiplications, and $N^T - 1$ additions.) This calculation is computationally infeasible, even for small values of $N$ and $T$; e.g., for $N = 5$ (states), $T = 100$ (observations), there are on the order of $2 \cdot 100 \cdot 5^{100} \approx 10^{72}$ computations! Clearly a more efficient procedure is required to solve problem 1. Fortunately such a procedure (called the forward procedure) exists.

#### 6.4.1.1 The Forward Procedure

Consider the forward variable $\alpha_t(i)$ defined as

$$\alpha_t(i) = P(\mathbf{o}_1 \mathbf{o}_2 \ldots \mathbf{o}_t, q_t = i|\lambda) \tag{6.18}$$

that is, the probability of the partial observation sequence, $\mathbf{o}_1\, \mathbf{o}_2 \ldots \mathbf{o}_t$, (until time $t$) and state $i$ at time $t$, given the model $\lambda$. We can solve for $\alpha_t(i)$ inductively, as follows:

1. Initialization

$$\alpha_1(i) = \pi_i b_i(\mathbf{o}_1), \qquad 1 \le i \le N. \tag{6.19}$$

2. Induction

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(\mathbf{o}_{t+1}), \qquad \begin{matrix} 1 \le t \le T-1 \\ 1 \le j \le N \end{matrix} \tag{6.20}$$

3. Termination

$$P(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i). \tag{6.21}$$

Step 1 initializes the forward probabilities as the joint probability of state $i$ and initial observation $\mathbf{o}_1$. The induction step, which is the heart of the forward calculation, is illustrated in Figure 6.5(a). This figure shows how state $j$ can be reached at time $t + 1$ from the $N$ possible states, $i$, $1 \le i \le N$, at time $t$. Since $\alpha_t(i)$ is the probability of the joint event that $\mathbf{o}_1\, \mathbf{o}_2 \ldots \mathbf{o}_t$ are observed, and the state at time $t$ is $i$, the product $\alpha_t(i) a_{ij}$ is

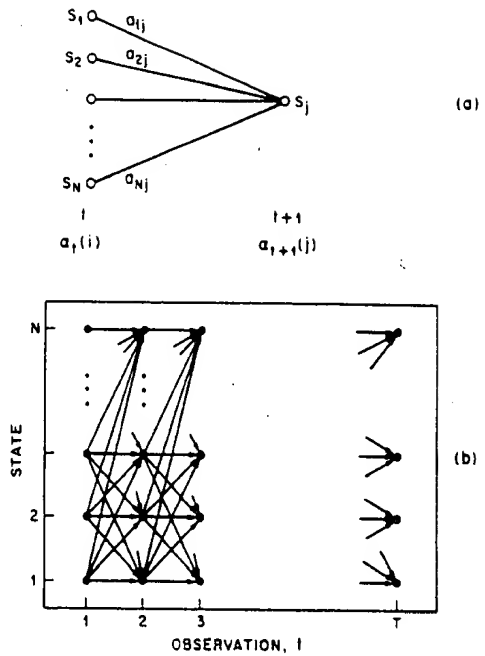



**Figure 6.5** (a) Illustration of the sequence of operations required for the computation of the forward variable $\alpha_{t+1}(j)$. (b) Implementation of the computation of $\alpha_t(i)$ in terms of a lattice of observations $t$, and states $i$.

then the probability of the joint event that $\mathbf{o}_1\,\mathbf{o}_2 \ldots \mathbf{o}_t$ are observed, and state $j$ is reached at time $t+1$ via state $i$ at time $t$. Summing this product over all the $N$ possible states, $i$, $1 \le i \le N$ at time $t$ results in the probability of $j$ at time $t+1$ with all the accompanying previous partial observations. Once this is done and $j$ is known, it is easy to see that $\alpha_{t+1}(j)$ is obtained by accounting for observation $\mathbf{o}_{t+1}$ in state $j$, i.e., by multiplying the summed quantity by the probability $b_j(\mathbf{o}_{t+1})$. The computation of Eq. (6.20) is performed for all states $j$, $1 \le j \le N$, for a given $t$; the computation is then iterated for $t = 1, 2, \ldots, T-1$. Finally, step 3 gives the desired calculation of $P(\mathbf{O}|\lambda)$ as the sum of the terminal forward variables $\alpha_T(i)$. This is the case since, by definition,

$$\alpha_T(i) = P(\mathbf{o}_1\mathbf{o}_2 \ldots \mathbf{o}_T, q_T = i|\lambda) \tag{6.22}$$

and hence $P(\mathbf{O}|\lambda)$ is just the sum of the $\alpha_T(i)$'s.

If we examine the computation involved in the calculation of $\alpha_t(j)$, $1 \le t \le T$, $1 \le j \le N$, we see that it requires on the order of $N^2T$ calculations, rather than $2TN^T$ as required by the direct calculation. (Again, to be precise, we need $N(N+1)(T-1)+N$ multiplications and $N(N-1)(T-1)$ additions.) For $N = 5$, $T = 100$, we need about 3000 computations for the forward method, versus $10^{72}$ computations for the direct calculation, a savings of about 69 orders of magnitude.



The forward probability calculation is, in effect, based upon the lattice (or trellis) structure shown in Figure 6.5(b). The key is that, because there are only $N$ states (nodes at each time slot in the lattice), all the possible state sequences will remerge into these $N$ nodes, no matter how long the observation sequence. At time $t = 1$ (the first time slot in the lattice), we need to calculate values of $\alpha_1(i)$, $1 \le i \le N$. At times $t = 2, 3, \ldots, T$, we need only calculate values of $\alpha_t(j)$, $1 \le j \le N$, where each calculation involves only the $N$ previous values of $\alpha_{t-1}(i)$ because each of the $N$ grid points can be reached from only the $N$ grid points at the previous time slot.

#### 6.4.1.2 The Backward Procedure

In a similar manner, we can consider a backward variable $\beta_t(i)$ defined as

$$\beta_t(i) = P(\mathbf{o}_{t+1}\mathbf{o}_{t+2} \ldots \mathbf{o}_T|q_t = i, \lambda) \tag{6.23}$$

that is, the probability of the partial observation sequence from $t+1$ to the end, given state $i$ at time $t$ and the model $\lambda$. Again we can solve for $\beta_t(i)$ inductively, as follows:

1. Initialization
$$\beta_T(i) = 1, \qquad 1 \le i \le N. \tag{6.24}$$
2. Induction
$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j),$$
$$t = T-1, T-2, \ldots, 1, \qquad 1 \le i \le N. \tag{6.25}$$

The initialization step 1 *arbitrarily* defines $\beta_T(i)$ to be 1 for all $i$. Step 2, which is illustrated in Figure 6.6, shows that in order to have been in state $i$ at time $t$, and to account for the observation sequence from time $t+1$ on, you have to consider all possible states $j$ at time $t+1$, accounting for the transition from $i$ to $j$ (the $a_{ij}$ term), as well as the observation $\mathbf{o}_{t+1}$ in state $j$ (the $b_j(\mathbf{o}_{t+1})$ term), and then account for the remaining partial observation sequence from state $j$ (the $\beta_{t+1}(j)$ term). We will see later how the backward as well as the forward calculations are used to help solve fundamental Problems 2 and 3 of HMMs.

Again, the computation of $\beta_t(i)$, $1 \le t \le T$, $1 \le i \le N$, requires on the order of $N^2T$ calculations, and can be computed in a lattice structure similar to that of Figure 6.5(b).

### 6.4.2 Solution to Problem 2—"Optimal" State Sequence

Unlike Problem 1, for which an exact solution can be given, there are several possible ways of solving Problem 2—namely, finding the "optimal" state sequence associated with the given observation sequence. The difficulty lies with the definition of the optimal state sequence—that is, there are several possible optimality criteria. For example, one possible

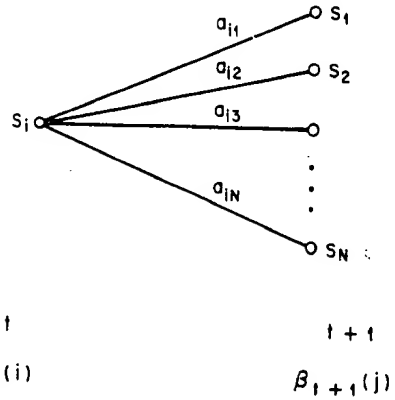



**Figure 6.6** Sequence of operations required for the computation of the backward variable $\beta_t(i)$.

optimality criterion is to choose the states $q_t$ that are *individually* most likely at each time $t$. This optimality criterion maximizes the expected number of correct individual states. To implement this solution to Problem 2, we can define the a posteriori probability variable

$$\gamma_t(i) = P(q_t = i|\mathbf{O}, \lambda) \tag{6.26}$$

that is, the probability of being in state $i$ at time $t$, given the observation sequence $\mathbf{O}$, and the model $\lambda$. We can express $\gamma_t(i)$ in several forms, including

$$\begin{aligned}\gamma_t(i) &= P(q_t = i \mid \mathbf{O}, \lambda) \\ &= \frac{P(\mathbf{O}, q_t = i \mid \lambda)}{P(\mathbf{O} \mid \lambda)} \\ &= \frac{P(\mathbf{O}, q_t = i \mid \lambda)}{\sum_{i=1}^{N} P(\mathbf{O}, q_t = i \mid \lambda)}.\end{aligned} \tag{6.27}$$

Since $P(\mathbf{O}, q_t = i \mid \lambda)$ is equal to $\alpha_t(i)\beta_t(i)$, we can write $\gamma_t(i)$ as

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)} \tag{6.28}$$

where we see that $\alpha_t(i)$ accounts for the partial observation sequence $\mathbf{o}_1\, \mathbf{o}_2 \ldots \mathbf{o}_t$ and state $i$ at $t$, while $\beta_t(i)$ accounts for the remainder of the observation sequence $\mathbf{o}_{t+1}\mathbf{o}_{t+2} \ldots \mathbf{o}_T$, given state $q_t = i$ at $t$.



Using $\gamma_t(i)$, we can solve for the individually most likely state $q_t^*$ at time $t$, as

$$q_t^* = \arg \min_{1 \le i \le N} [\gamma_t(i)], \qquad 1 \le t \le T. \tag{6.29}$$

Although Eq. (6.29) maximizes the expected number of correct states (by choosing the most likely state for each $t$), there could be some problems with the resulting state sequence. For example, when the HMM has state transitions which have zero probability ($a_{ij} = 0$ for some $i$ and $j$), the "optimal" state sequence may, in fact, not even be a valid state sequence. This is because the solution of Eq. (6.29) simply determines the most likely state at every instant, without regard to the probability of occurrence of sequences of states.

One possible solution to the above problem is to modify the optimality criterion. For example, one could solve for the state sequence that maximizes the expected number of correct pairs of states $(q_t, q_{t+1})$, or triples of states $(q_t, q_{t+1}, q_{t+2})$, etc. Although these criteria might be reasonable for some applications, the most widely used criterion is to find the *single* best state sequence (path)—that is, to maximize $P(\mathbf{q}|\mathbf{O}, \lambda)$, which is equivalent to maximizing $P(\mathbf{q}, \mathbf{O}|\lambda)$. A formal technique for finding this single best state sequence exists, based on dynamic programming methods, and is called the Viterbi algorithm [15, 16].

#### 6.4.2.1 The Viterbi Algorithm

To find the single best state sequence, $\mathbf{q} = (q_1\, q_2 \ldots q_T)$, for the given observation sequence $\mathbf{O} = (\mathbf{o}_1\, \mathbf{o}_2 \ldots \mathbf{o}_T)$, we need to define the quantity

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P[q_1 q_2 \ldots q_{t-1},\ q_t = i,\ \mathbf{o}_1 \mathbf{o}_2 \ldots \mathbf{o}_t|\lambda] \tag{6.30}$$

that is, $\delta_t(i)$ is the best score (highest probability) along a single path, at time $t$, which accounts for the first $t$ observations and ends in state $i$. By induction we have

$$\delta_{t+1}(j) = [\max_i \delta_t(i)\, a_{ij}] \cdot b_j(\mathbf{o}_{t+1}). \tag{6.31}$$

To actually retrieve the state sequence, we need to keep track of the argument that maximized Eq. (6.31), for each $t$ and $j$. We do this via the array $\psi_t(j)$. The complete procedure for finding the best state sequence can now be stated as follows:

1. Initialization

$$\delta_1(i) = \pi_i b_i(\mathbf{o}_1), \qquad 1 \le i \le N \tag{6.32a}$$

$$\psi_1(i) = 0. \tag{6.32b}$$

2. Recursion

$$\delta_t(j) = \max_{1 \le i \le N} [\delta_{t-1}(i)\, a_{ij}] b_j(\mathbf{o}_t), \qquad \begin{matrix} 2 \le t \le T \\ 1 \le j \le N \end{matrix} \tag{6.33a}$$

$$\psi_t(j) = \arg \max_{1 \le i \le N} [\delta_{t-1}(i)\, a_{ij}], \qquad \begin{matrix} 2 \le t \le T \\ 1 \le j \le N. \end{matrix} \tag{6.33b}$$

3. Termination

$$P^* = \max_{1 \le i \le N} [\delta_T(i)] \tag{6.34a}$$

$$q_T^* = \arg \max_{1 \le i \le N} [\delta_T(i)]. \tag{6.34b}$$

4. Path (state sequence) backtracking

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \qquad t = T-1, T-2, \ldots, 1. \tag{6.35}$$

It should be noted that the Viterbi algorithm is similar (except for the backtracking step) in implementation to the forward calculation of Eqs. (6.19)–(6.21). The major difference is the maximization in Eq. (6.33a) over previous states, which is used in place of the summing procedure in Eq. (6.20). It also should be clear that a lattice (or trellis) structure efficiently implements the computation of the Viterbi procedure.

### 6.4.2.2 Alternative Viterbi Implementation

By taking logarithms of the model parameters, the Viterbi algorithm of the preceding section can be implemented without the need for any multiplications. Thus:

**0. Preprocessing**

$$\begin{aligned} \tilde{\pi}_i &= \log(\pi_i), && 1 \le i \le N \\ \tilde{b}_i(\mathbf{o}_t) &= \log[b_i(\mathbf{o}_t)], && 1 \le i \le N,\ 1 \le t \le T \\ \tilde{a}_{ij} &= \log(a_{ij}), && 1 \le i, j \le N \end{aligned}$$

**1. Initialization**

$$\begin{aligned} \tilde{\delta}_1(i) &= \log(\delta_1(i)) = \tilde{\pi}_i + \tilde{b}_i(\mathbf{o}_1), && 1 \le i \le N \\ \psi_1(i) &= 0, && 1 \le i \le N \end{aligned}$$

**2. Recursion**

$$\tilde{\delta}_t(j) = \log(\delta_t(j)) = \max_{1 \le i \le N} \left[\tilde{\delta}_{t-1}(i) + \tilde{a}_{ij}\right] + \tilde{b}_j(\mathbf{o}_t)$$

$$\psi_t(j) = \arg \max_{1 \le i \le N} [\tilde{\delta}_{t-1}(i) + \tilde{a}_{ij}], \qquad 2 \le t \le T,\ 1 \le j \le N$$

**3. Termination**

$$\tilde{P}^* = \max_{1 \le i \le N} [\tilde{\delta}_T(i)]$$

$$q_T^* = \arg \max_{1 \le i \le N} [\tilde{\delta}_T(i)]$$

**4. Backtracking**

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \qquad t = T-1, T-2, \ldots, 1$$

The calculation required for this alternative implementation is on the order of $N^2T$ additions (plus the calculation for preprocessing). Because the preprocessing needs to be performed once and saved, its cost is negligible for most systems.



**Exercise 6.3**

Given the model of the coin-toss experiment used in Exercise 6.2 (i.e., three different coins) with probabilities

| | State 1 | State 2 | State 3 |
|---|---|---|---|
| $P(H)$ | 0.5 | 0.75 | 0.25 |
| $P(T)$ | 0.5 | 0.25 | 0.75 |

and with all state transition probabilities equal to 1/3, and with initial probabilities equal to 1/3, for the observation sequence

$$O = (H\,H\,H\,H\,T\,H\,T\,T\,T\,T)$$

find the most likely path with the Viterbi algorithm.

**Solution 6.3**

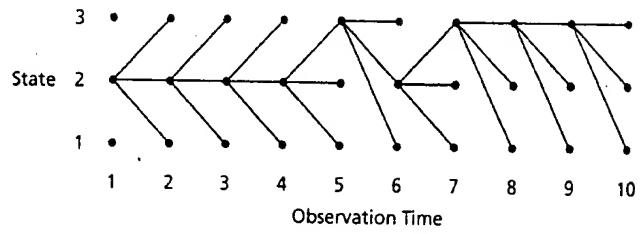
Since all $a_{ij}$ terms are equal to 1/3, we can omit these terms (as well as the initial state probability term), giving

$$\delta_1(1) = 0.5, \quad \delta_1(2) = 0.75, \quad \delta_1(3) = 0.25.$$

The recursion for $\delta_t(j)$ gives $(2 \le t \le 10)$

$$\begin{aligned} &\delta_2(1) = (0.75)(0.5), && \delta_2(2) = (0.75)^2, && \delta_2(3) = (0.75)(0.25) \\ &\delta_3(1) = (0.75)^2(0.5), && \delta_3(2) = (0.75)^3, && \delta_3(3) = (0.75)^2(0.25) \\ &\delta_4(1) = (0.75)^3(0.5), && \delta_4(2) = (0.75)^4, && \delta_4(3) = (0.75)^3(0.25) \\ &\delta_5(1) = (0.75)^4(0.5), && \delta_5(2) = (0.75)^4(0.25), && \delta_5(3) = (0.75)^5 \\ &\delta_6(1) = (0.75)^5(0.5), && \delta_6(2) = (0.75)^6, && \delta_6(3) = (0.75)^5(0.25) \\ &\delta_7(1) = (0.75)^6(0.5), && \delta_7(2) = (0.75)^6(0.25), && \delta_7(3) = (0.75)^7 \\ &\delta_8(1) = (0.75)^7(0.5), && \delta_8(2) = (0.75)^7(0.25), && \delta_8(3) = (0.75)^8 \\ &\delta_9(1) = (0.75)^8(0.5), && \delta_9(2) = (0.75)^8(0.25), && \delta_9(3) = (0.75)^9 \\ &\delta_{10}(1) = (0.75)^9(0.5), && \delta_{10}(2) = (0.75)^9(0.25), && \delta_{10}(3) = (0.75)^{10} \end{aligned}$$

This leads to a diagram (trellis) of the form:




Hence, the most likely state sequence is $\{2, 2, 2, 2, 3, 2, 3, 3, 3, 3\}$.

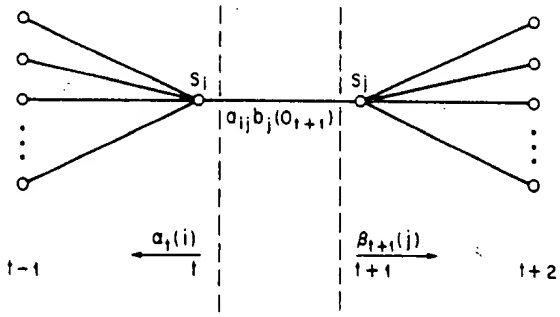



**Figure 6.7** Illustration of the sequence of operations required for the computation of the joint event that the system is in state $i$ at time $t$ and state $j$ at time $t+1$.

### 6.4.3 Solution to Problem 3—Parameter Estimation

The third, and by far the most difficult, problem of HMMs is to determine a method to adjust the model parameters $(A, B, \pi)$ to satisfy a certain optimization criterion. There is no known way to analytically solve for the model parameter set that maximizes the probability of the observation sequence in a closed form. We can, however, choose $\lambda = (A, B, \pi)$ such that its likelihood, $P(\mathbf{O}|\lambda)$, is locally maximized using an iterative procedure such as the Baum-Welch method (also known as the EM (expectation-maximization) method [17]), or using gradient techniques [18]. In this section we discuss one iterative procedure, based primarily on the classic work of Baum and his colleagues, for choosing the maximum likelihood (ML) model parameters.

To describe the procedure for reestimation (iterative update and improvement) of HMM parameters, we first define $\xi_t(i,j)$, the probability of being in state $i$ at time $t$, and state $j$ at time $t+1$, given the model and the observation sequence, i.e.

$$\xi_t(i,j) = P(q_t = i,\ q_{t+1} = j|\mathbf{O}, \lambda). \tag{6.36}$$

The paths that satisfy the conditions required by Eq. (6.36) are illustrated in Figure 6.7. From the definitions of the forward and backward variables, we can write $\xi_t(i,j)$ in the form

$$\begin{aligned}\xi_t(i,j) &= \frac{P(q_t = i,\ q_{t+1} = j,\ \mathbf{O} \mid \lambda)}{P(\mathbf{O} \mid \lambda)} \\ &= \frac{\alpha_t(i)\, a_{ij} b_j(\mathbf{o}_{t+1})\beta_{t+1}(j)}{P(\mathbf{O}|\lambda)} \\ &= \frac{\alpha_t(i)\, a_{ij} b_j(\mathbf{o}_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)\, a_{ij} b_j(\mathbf{o}_{t+1})\beta_{t+1}(j)}.\end{aligned} \tag{6.37}$$

We have previously defined $\gamma_t(i)$ as the probability of being in state $i$ at time $t$, given the entire observation sequence and the model; hence, we can relate $\gamma_t(i)$ to $\xi_t(i,j)$ by summing over $j$, giving

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j). \tag{6.38}$$

If we sum $\gamma_t(i)$ over the time index $t$, we get a quantity that can be interpreted as the expected (over time) number of times that state $i$ is visited, or equivalently, the expected number of transitions made from state $i$ (if we exclude the time slot $t = T$ from the summation). Similarly, summation of $\xi_t(i,j)$ over $t$ (from $t = 1$ to $t = T-1$) can be interpreted as the expected number of transitions from state $i$ to state $j$. That is,

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{expected number of transitions from state } i \text{ in } \mathbf{O} \tag{6.39a}$$

$$\sum_{t=1}^{T-1} \xi_t(i,j) = \text{expected number of transitions from state } i \text{ to state } j \text{ in } \mathbf{O}. \tag{6.39b}$$

Using the above formulas (and the concept of counting event occurrences), we can give a method for reestimation of the parameters of an HMM. A set of reasonable reestimation formulas for $\pi$, $A$, and $B$ is

$$\bar{\pi}_i = \text{expected frequency (number of times) in state } i \text{ at time } (t=1) = \gamma_1(i) \tag{6.40a}$$

$$\begin{aligned}\bar{a}_{ij} &= \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i} \\ &= \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}\end{aligned} \tag{6.40b}$$

$$\begin{aligned}\bar{b}_j(k) &= \frac{\text{expected number of times in state } j \text{ and observing symbol } \mathbf{v}_k}{\text{expected number of times in state } j} \\ &= \frac{\sum_{\substack{t=1 \\ \text{s.t. } \mathbf{o}_t = \mathbf{v}_k}}^{T}\gamma_t(j)}{\sum_{t=1}^{T}\gamma_t(j)}.\end{aligned} \tag{6.40c}$$

If we define the current model as $\lambda = (A, B, \pi)$ and use that to compute the right-hand sides of Eqs. (6.40a)–(6.40c), and we define the reestimated model as $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$, as determined from the left-hand sides of Eqs. (6.40a)–(6.40c), then it has been proven by Baum and his colleagues that either (1) the initial model $\lambda$ defines a critical point of the

likelihood function, in which case $\bar{\lambda} = \lambda$; or (2) model $\bar{\lambda}$ is more likely than model $\lambda$ in the sense that $P(\mathbf{O}|\bar{\lambda}) > P(\mathbf{O}|\lambda)$; that is, we have found a new model $\bar{\lambda}$ from which the observation sequence is more likely to have been produced.

Based on the above procedure, if we iteratively use $\bar{\lambda}$ in place of $\lambda$ and repeat the reestimation calculation, we then can improve the probability of $\mathbf{O}$ being observed from the model until some limiting point is reached. The final result of this reestimation procedure is an ML estimate of the HMM. It should be pointed out that the forward-backward algorithm leads to local maxima only, and that in most problems of interest, the likelihood function is very complex and has many local maxima.

The reestimation formulas of Eqs. (6.40a)–(6.40c) can be derived directly by maximizing (using standard constrained optimization techniques) Baum's auxiliary function

$$Q(\lambda', \lambda) = \sum_{\mathbf{q}} P(\mathbf{O}, \mathbf{q}|\lambda') \log\ P(\mathbf{O}, \mathbf{q}|\lambda) \tag{6.41}$$

over $\lambda$. Because

$$Q(\lambda', \lambda) \geq Q(\lambda', \lambda') \Rightarrow P(\mathbf{O}|\lambda) \geq P(\mathbf{O}|\lambda') \tag{6.42}$$

we can maximize the function $Q(\lambda', \lambda)$ over $\lambda$ to improve $\lambda'$ in the sense of increasing the likelihood $P(\mathbf{O}|\lambda)$. Eventually the likelihood function converges to a critical point if we iterate the procedure.

### 6.4.3.1 Derivation of Reestimation Formulas from the $Q$ Function

The auxiliary function $Q(\lambda', \lambda)$ was defined in Eq. (6.41) as

$$Q(\lambda', \lambda) = \sum_{\mathbf{q}} P(\mathbf{O}, \mathbf{q}|\lambda') \log\ P(\mathbf{O}, \mathbf{q}|\lambda)$$

in which we can express $P$ and $\log\ P$ (in terms of the HMM parameters) as

$$P(\mathbf{O}, \mathbf{q}|\lambda) = \pi_{q_0} \prod_{t=1}^{T} a_{q_{t-1}q_t} b_{q_t}(\mathbf{o}_t)$$

$$\log\ P(\mathbf{O}, \mathbf{q}|\lambda) = \log\ \pi_{q_0} + \sum_{t=1}^{T} \log\ a_{q_{t-1}q_t} + \sum_{t=1}^{T} \log\ b_{q_t}(\mathbf{o}_t)$$

(There is a slight difference between the above equations and the expression of Eq. (6.17) in which the first observation is associated with the initial state before any state transition is made. This difference is inconsequential and should not impede our understanding of the method.) Thus we can write $Q(\lambda', \lambda)$ as

$$Q(\lambda', \lambda) = Q_\pi(\lambda', \pi) + \sum_{i=1}^{N} Q_{a_i}(\lambda', \mathbf{a}_i) + \sum_{i=1}^{N} Q_{b_i}(\lambda', \mathbf{b}_i)$$

where

$$\pi = [\pi_1, \pi_2, \ldots, \pi_N],$$



$$\mathbf{a}_i = [a_{i1}, a_{i2}, \ldots, a_{iN}], \quad \mathbf{b}_i \text{ is the parameter vector that defines } b_i(\cdot)$$

and

$$Q_\pi(\lambda', \pi) = \sum_{i=1}^{N} P(\mathbf{O}, q_0 = i|\lambda') \log\ \pi_i$$

$$Q_{a_i}(\lambda', \mathbf{a}_i) = \sum_{j=1}^{N} \sum_{t=1}^{T} P(\mathbf{O}, q_{t-1} = i, q_t = j|\lambda') \log\ a_{ij}$$

$$Q_{b_i}(\lambda', \mathbf{b}_i) = \sum_{t=1}^{T} P(\mathbf{O}, q_t = i|\lambda') \log\ b_i(\mathbf{o}_t)$$

Because of the separability of $Q(\lambda', \lambda)$ into three independent terms, we can maximize $Q(\lambda', \lambda)$ over $\lambda$ by maximizing the individual terms separately, subject to the stochastic constraints

$$\sum_{j=1}^{N} \pi_j = 1$$

$$\sum_{j=1}^{N} a_{ij} = 1, \qquad \forall j$$

and (for discrete densities where $b_i(\mathbf{o}_t = \mathbf{v}_k) = b_i(k)$)

$$\sum_{k=1}^{K} b_i(k) = 1, \qquad \forall i.$$

Because the individual auxiliary functions all have the form

$$\sum_{j=1}^{N} w_j \log\ y_j$$

which, as a function of $\{y_j\}_{j=1}^{N}$, subject to the constraints $\sum_{j=1}^{N} y_j = 1$, $y_j \geq 0$, attains a global maximum at the single point

$$y_j = \frac{w_j}{\sum_{i=1}^{N} w_i}, \qquad j = 1, 2, \ldots, N$$

then the maximization leads to the model reestimate $\bar{\lambda} = [\bar{\pi}, \bar{A}, \bar{B}]$ where

$$\bar{\pi}_i = \frac{P(\mathbf{O}, q_0 = i|\lambda)}{P(\mathbf{O}|\lambda)}$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T} P(\mathbf{O}, q_{t-1} = i, q_t = j|\lambda)}{\sum_{t=1}^{T} P(\mathbf{O}, q_{t-1} = i|\lambda)}$$

$$\bar{b}_i(k) = \frac{\sum_{t=1}^{T} P(\mathbf{O}, q_t = i|\lambda)\ \delta(\mathbf{o}_t, \mathbf{v}_k)}{\sum_{t=1}^{T} P(\mathbf{O},\ q_t = i|\lambda)}$$

where we have defined

$$\delta(\mathbf{o}_t, \mathbf{v}_k) = 1 \quad \text{if } \mathbf{o}_t = \mathbf{v}_k$$
$$= 0 \quad \text{otherwise.}$$

Using the definitions of the forward variable, $\alpha_t(i) = P(\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_t, q_t = i|\lambda)$ and the backward variable, $\beta_t(i) = P(\mathbf{o}_{t+1}, \ldots, \mathbf{o}_T|q_t = i, \lambda)$, the reestimation transformations can be easily calculated as

$$P(\mathbf{O}, q_t = i|\lambda) = \alpha_t(i)\beta_t(i)$$

$$P(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i) = \sum_{i=1}^{N} \alpha_T(i)$$

$$P(\mathbf{O}, q_{t-1} = i,\ q_t = j|\lambda) = \alpha_{t-1}(i)\, a_{ij} b_j(\mathbf{o}_t)\beta_t(j)$$

giving

$$\bar{\pi}_i = \frac{\alpha_0(i)\beta_0(i)}{\sum_{j=1}^{N} \alpha_T(j)} = \gamma_0(i)$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T} \alpha_{t-1}(i)\, a_{ij} b_j(\mathbf{o}_t)\beta_t(j)}{\sum_{t=1}^{T} \alpha_{t-1}(i)\beta_{t-1}(i)} = \frac{\sum_{t=1}^{T} \xi_{t-1}(i,j)}{\sum_{t=1}^{T} \gamma_{t-1}(i)}$$

$$\bar{b}_i(k) = \frac{\sum_{t=1}^{T} \alpha_t(i)\beta_t(i)\delta(\mathbf{o}_t, \mathbf{v}_k)}{\sum_{t=1}^{T} \alpha_t(i)\beta_t(i)} = \frac{\sum_{\substack{t=1 \\ \text{s.t. } \mathbf{o}_t = \mathbf{v}_k}}^{T} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)}$$

which are the formulas given in Eqs. (6.40a)–(6.40c).



### 6.4.4 Notes on the Reestimation Procedure

The reestimation formulas can be readily interpreted as an implementation of the EM algorithm of statistics [17] in which the E (expectation) step is the calculation of the auxiliary function $Q(\lambda', \lambda)$, (which is the expectation of log $P(\mathbf{O}, \mathbf{q}|\lambda)$), and the M (maximization) step is the maximization of $Q(\lambda', \lambda)$ over $\lambda$ to obtain $\bar{\lambda}$. Thus the Baum-Welch reestimation equations are essentially identical to the EM steps for this particular problem.

An important property of the reestimation procedure is that the stochastic constraints of the HMM parameters, namely

$$\sum_{i=1}^{N} \bar{\pi}_i = 1 \tag{6.43a}$$

$$\sum_{j=1}^{N} \bar{a}_{ij} = 1, \qquad 1 \le i \le N \tag{6.43b}$$

$$\sum_{k=1}^{M} \bar{b}_j(k) = 1, \qquad 1 \le j \le N \tag{6.43c}$$

are automatically incorporated at each iteration. By looking at the parameter estimation problem as a constrained optimization of $P(\mathbf{O}|\lambda)$ (subject to the constraints of Eq. (6.43)), we can formulate the solution procedure by use of the techniques of variational calculus to maximize $P$ (we use the notation $P = P(\mathbf{O}|\lambda)$ as shorthand in this section). Based on a standard Lagrange optimization setup using Lagrange multipliers, it can readily be shown that $P$ is maximized when the following conditions are met:

$$\pi_i = \frac{\pi_i \dfrac{\partial P}{\partial \pi_i}}{\sum_{k=1}^{N} \pi_k \dfrac{\partial P}{\partial \pi_k}} \tag{6.44a}$$

$$a_{ij} = \frac{a_{ij} \dfrac{\partial P}{\partial a_{ij}}}{\sum_{k=1}^{N} a_{ik} \dfrac{\partial P}{\partial a_{ik}}} \tag{6.44b}$$

$$b_j(k) = \frac{b_j(k) \dfrac{\partial P}{\partial b_j(k)}}{\sum_{\ell=1}^{M} b_j(\ell) \dfrac{\partial P}{\partial b_j(\ell)}}. \tag{6.44c}$$

By appropriate manipulation of Eq. (6.44), the right-hand sides of each equation can be readily shown to be *identical* to the right-hand sides of each part of Eqs. (6.40a)–(6.40c), thereby showing that the reestimation formulas are indeed exactly correct at critical points of $P$. In fact, the form of Eq. (6.44) is essentially that of a reestimation formula in which

the left-hand side is the reestimate and the right-hand side is computed using the current values of the variables.

Finally, we note that since the entire problem can be set up as an optimization problem, standard gradient techniques can be used to solve for "optimal" values of the model parameters. Such procedures have been tried and have been shown to yield solutions comparable to those of the standard reestimation procedures [18]. One critical shortcoming of standard gradient technique, as applied to the maximization of $P(\mathbf{O}|\lambda)$, is that the descent algorithms, which are critically dependent on taking a small step in the direction of the gradient, often do not produce *monotonic* improvement in the likelihood as the Baum-Welch reestimation is guaranteed by Eq. (6.42) to do.

## 6.5 TYPES OF HMMS

One way to classify types of HMMs is by the structure of the transition matrix, $A$, of the Markov chain. Until now, we have only considered the special case of ergodic or fully connected HMMs in which every state of the model could be reached (in a single step) from every other state of the model. (Strictly speaking, an ergodic model has the property that every state can be reached from every other state in a finite but aperiodic number of steps.) As shown in Figure 6.8(a), for an $N = 4$ state model, this type of model has the property that every $a_{ij}$ coefficient is positive. Hence for the example of Figure 6.8(a) we have

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}.$$

For some applications, particularly those to be discussed later in this chapter, other types of HMMs have been found to account for observed properties of the signal being modeled better than the standard ergodic model. One such model is shown in Figure 6.8(b). This model is called a left-right model or a Bakis model ([11], [10]) because the underlying state sequence associated with the model has the property that, as time increases, the state index increases (or stays the same)—that is, the system states proceed from left to right. Clearly the left-right type of HMM has the desirable property that it can readily model signals whose properties change over time in a successive manner—e.g., speech. The fundamental property of all left-right HMMs is that the state-transition coefficients have the property

$$a_{ij} = 0, \qquad j < i \tag{6.45}$$

that is, no transitions are allowed to states whose indices are lower than that of the current state. Furthermore, the initial state probabilities have the property

$$\pi_i = \begin{cases} 0, & i \neq 1 \\ 1, & i = 1 \end{cases} \tag{6.46}$$

because the state sequence must begin in state 1 (and end in state $N$). Often, with left-right



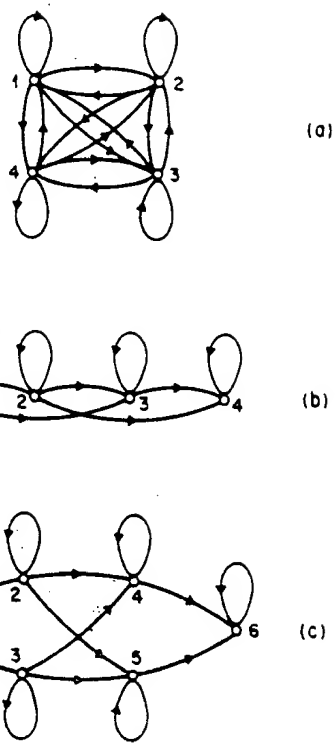



**Figure 6.8** Illustration of three distinct types of HMMs. (a) A 4-state ergodic model. (b) A 4-state left-right model. (c) A 6-state parallel path left-right model.

models, additional constraints are placed on the state-transition coefficients to make sure that large changes in state indices do not occur; hence a constraint of the form

$$a_{ij} = 0, \qquad j > i + \Delta i \tag{6.47}$$

is often used. In particular, for the example of Figure 6.8(b), the value of $\Delta i$ is 2; that is, no jumps of more than two states are allowed. The form of the state-transition matrix for the example of Figure 6.8(b) is thus

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix}.$$

It should be clear that, for the last state in a left-right model, the state-transition coefficients are specified as

$$a_{NN} = 1 \tag{6.48a}$$

$$a_{Ni} = 0, \qquad i < N. \tag{6.48b}$$

Besides the above fully connected and left-right models, there are many other possible variations and combinations. By way of example, Figure 6.8(c) shows a cross-coupled connection of two parallel left-right HMMs. Strictly speaking, this model is a left-right model (it obeys all the $a_{ij}$ constraints); however, it has certain flexibility not present in a strict left-right model (i.e., one without parallel paths).

It should be clear that the imposition of the constraints of the left-right model, or those of the constrained jump model, essentially have no effect on the reestimation procedure. This is the case because any HMM parameter set to zero initially will remain at zero throughout the reestimation procedure (see Eq. (6.44)).

## 6.6 CONTINUOUS OBSERVATION DENSITIES IN HMMS

All of our discussion to this point has considered only when the observations were characterized as discrete symbols chosen from a finite alphabet, and therefore we could use a discrete probability density within each state of this model ([19]–[21]). The problem with this approach, at least for some applications, is that the observations are often continuous signals (or vectors). Although it is possible to convert such continuous signal representations into a sequence of discrete symbols via vector quantization codebooks and other methods, there might be serious degradation associated with such discretization of the continuous signal. Hence it would be advantageous to be able to use HMMs with continuous observation densities to model continuous signal representations directly.

To use a continuous observation density, some restrictions must be placed on the form of the model probability density function (pdf) to ensure that the parameters of the pdf can be reestimated in a consistent way. The most general representation of the pdf, for which a reestimation procedure has been formulated, is a finite mixture of the form

$$b_j(\mathbf{o}) = \sum_{k=1}^{M} c_{jk} \mathcal{N}(\mathbf{o}, \boldsymbol{\mu}_{jk}, \mathbf{U}_{jk}), \qquad 1 \le j \le N \tag{6.49}$$

where $\mathbf{o}$ is the observation vector being modeled, $c_{jk}$ is the mixture coefficient for the $k$th mixture in state $j$ and $\mathcal{N}$ is any log-concave or elliptically symmetric density [18] (e.g., Gaussian). Without loss of generality, we assume that $\mathcal{N}$ is Gaussian in Eq. (6.49) with mean vector $\boldsymbol{\mu}_{jk}$ and covariance matrix $\mathbf{U}_{jk}$ for the $k$th mixture component in state $j$. The mixture gains $c_{jk}$ satisfy the stochastic constraint

$$\sum_{k=1}^{M} c_{jk} = 1, \qquad 1 \le j \le N \tag{6.50a}$$

$$c_{jk} \ge 0, \; 1 \le j \le N, \; 1 \le k \le M \tag{6.50b}$$



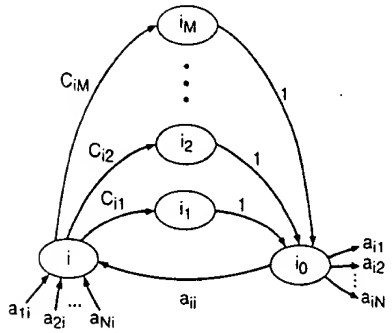



**Figure 6.9** Equivalence of a state with a mixture density to a multistate single-density distribution (after Juang et al. [21]).

so that the pdf is properly normalized, i.e.,

$$\int_{-\infty}^{\infty} b_j(\mathbf{o}) d\mathbf{o} = 1, \qquad 1 \le j \le N. \tag{6.51}$$

The pdf of Eq. (6.49) can be used to approximate, arbitrarily closely, any finite, continuous-density function. Hence it can be applied to a wide range of problems.

It has been shown that an HMM state with a mixture density is equivalent to a multistate single-mixture density model in the following way [21]. Consider a state $i$ with an $M$-mixture Gaussian density. Because the mixture gain coefficients sum up to 1, they define a set of transition coefficients to substates $i_1$ (with transition probability $c_{i1}$), $i_2$ (with transition probability $c_{i2}$) through $i_M$ (with transition probability $c_{iM}$). Within each substate $i_k$, there is a single mixture with mean $\boldsymbol{\mu}_{ik}$ and variance $\mathbf{U}_{ik}$ (see Figure 6.9 for a graphical interpretation). Each substate makes a transition to a wait state $i_0$ with probability 1. The distribution of the composite set of substates (each with a single density) is mathematically equivalent to the composite mixture density within a single state.

It can be shown that the reestimation formulas for the coefficients of the mixture density, i.e., $c_{jk}$, $\boldsymbol{\mu}_{jk}$, and $\mathbf{U}_{jk}$, are of the form

$$\bar{c}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)}{\sum_{t=1}^{T} \sum_{k=1}^{M} \gamma_t(j,k)} \tag{6.52}$$

$$\bar{\boldsymbol{\mu}}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k) \cdot \mathbf{o}_t}{\sum_{t=1}^{T} \gamma_t(j,k)} \tag{6.53}$$

$$\bar{\mathbf{U}}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k) \cdot (\mathbf{o}_t - \boldsymbol{\mu}_{jk})(\mathbf{o}_t - \boldsymbol{\mu}_{jk})'}{\sum_{t=1}^{T} \gamma_t(j,k)} \tag{6.54}$$

where prime denotes vector transpose and where $\gamma_t(j,k)$ is the probability of being in state $j$ at time $t$ with the $k$th mixture component accounting for $\mathbf{o}_t$, i.e.,

$$\gamma_t(j,k) = \left[ \frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} \right] \left[ \frac{c_{jk}\mathcal{N}(\mathbf{o}_t, \boldsymbol{\mu}_{jk}, \mathbf{U}_{jk})}{\sum_{m=1}^{M} c_{jm}\mathcal{N}(\mathbf{o}_t, \boldsymbol{\mu}_{jm}, \mathbf{U}_{jm})} \right].$$

(The term $\gamma_t(j,k)$ generalizes to $\gamma_t(j)$ of Eq. (6.26) in the case of a simple mixture, or a discrete density.) The reestimation formula for $a_{ij}$ is identical to the one used for discrete observation densities (i.e., Eq. (6.40(b))). The interpretation of Eqs. (6.52)–(6.54) is fairly straightforward. The reestimation formula for $c_{jk}$ is the ratio between the expected number of times the system is in state $j$ using the $k$th mixture component, and the expected number of times the system is in state $j$. Similarly, the reestimation formula for the mean vector $\boldsymbol{\mu}_{jk}$ weights each numerator term of Eq. (6.52) by the observation, thereby giving the expected value of the portion of the observation vector accounted for by the $k$th mixture component. A similar interpretation can be given for the reestimation term for the covariance matrix $\mathbf{U}_{jk}$.

## 6.7 AUTOREGRESSIVE HMMS

Another very interesting class of HMMs that is particularly applicable to speech processing is the class of autoregressive HMMs ([22, 23]). For this class, the observation vectors are drawn from an autoregression process. (The autoregressive density is, of course, just another continuous-probability density. However, we elaborate on the subject here separately from Section 6.6 because of its importance in speech analysis as will be shown later.)

To be more specific, consider the observation vector $\mathbf{o} = (x_0, x_1, x_2, \ldots, x_{K-1})$. The elements, $x_i$, could be simply the speech waveform samples. The components of $\mathbf{o}$ are assumed to be from an autoregressive Gaussian source, satisfying the relationship

$$x_k = -\sum_{i=1}^{p} a_i x_{k-i} + e_k \tag{6.55}$$

where $e_k$, $k = 0, 1, 2, \ldots, K-1$ are Gaussian, independent, identically distributed random variables with zero mean and variance $\sigma_e^2$, and $a_i$, $i = 1, 2, \ldots, p$, are the autoregression or predictor coefficients. It can be shown that for large $K$ [22, 23], the density function for $\mathbf{o}$



is approximately

$$f(\mathbf{o}) = (2\pi\sigma_e^2)^{-K/2} \exp\left\{ -\frac{1}{2\sigma_e^2}\delta(\mathbf{o}, \mathbf{a}) \right\} \tag{6.56}$$

where

$$\delta(\mathbf{o}, \mathbf{a}) = r_a(0)r(0) + 2\sum_{i=1}^{p} r_a(i)r(i) \tag{6.57a}$$

$$\mathbf{a} = [1, a_1, a_2, \ldots, a_p]', \qquad (a_0 = 1) \tag{6.57b}$$

$$r_a(i) = \sum_{n=0}^{p-i} a_n a_{n+i}, \qquad 1 \le i \le p \tag{6.57c}$$

$$r(i) = \sum_{n=0}^{K-i-1} x_n x_{n+i} \qquad 0 \le i \le p. \tag{6.57d}$$

In the above equations $r(i)$ is the autocorrelation of the observation samples, and $r_a(i)$ is the autocorrelation of the autoregressive coefficients. Furthermore, $\delta(\mathbf{o}, \mathbf{a})$ is a form of residual energy resulting from inverse filtering the data $x_i$ with an all-zero filter defined by $\mathbf{a}$. (See Eqs. 4.40–44.)

As discussed in Chapter 4, the signal level in the observation $\mathbf{o}$ is often treated in a different fashion from the general spectral shape when it comes to speech-pattern comparison. One way to separate the signal level from the spectral shape is to use gain normalization; that is, we use $\hat{\mathbf{o}}$ instead of $\mathbf{o}$ as the observation, where

$$\hat{\mathbf{o}} = \mathbf{o}/\sigma_{e\mathbf{o}} \tag{6.58}$$

and where $\sigma_{e\mathbf{o}}^2$ is the minimum linear prediction residual energy per sample. (It is shown in Exercise 6.5 that $\sigma_{e\mathbf{o}}^2 = (\sigma_e^2)_{\mathrm{ML}}$ for *the given observation* $\mathbf{o}$.) The elements of $\hat{\mathbf{o}}$, $\hat{x}_k = x_k/\sigma_{e\mathbf{o}}$, still satisfy the autoregressive relationship

$$\hat{x}_k = -\sum_{i=1}^{p} a_i \hat{x}_{k-i} + \hat{e}_k. \tag{6.59}$$

However, the variance of $\hat{e}_k$ is now unity. Therefore, we can write the probability density function for the output of an all-pole system defined by $\mathbf{a}$, driven by a zero mean, unit variance Gaussian i.i.d. sequence, as

$$f(\hat{\mathbf{o}}) = (2\pi)^{-K/2} \exp\left\{ -\frac{1}{2}\delta(\hat{\mathbf{o}}, \mathbf{a}) \right\} \tag{6.60}$$

if the data dimension $K$ is sufficiently large. (Note that the normalization factor $\sigma_{e\mathbf{o}}$ depends on the original data observation $\mathbf{o}$.) This type of pdf is often referred to as a "gain-independent" pdf.

The way in which we use a Gaussian autoregressive density in HMMs is straightfor-

ward. We assume a mixture density of the form

$$b_j(\mathbf{o}) = \sum_{k=1}^{M} c_{jk} b_{jk}(\mathbf{o}) \tag{6.61}$$

where each $b_{jk}(\mathbf{o})$ is the density defined by Eq. (6.60) with autoregression vector $\mathbf{a}_{jk}$ (or equivalently by autocorrelation vector $\mathbf{r}_{a_{jk}}$); that is,

$$b_{jk}(\mathbf{o}) = (2\pi)^{-K/2} \exp\left\{-\frac{1}{2}\delta(\mathbf{o}, \mathbf{a}_{jk})\right\}. \tag{6.62}$$

A reestimation formula for the sequence autocorrelation for the $j$th state, $k$th mixture component has been derived [22, 23], and is of the form

$$\bar{\mathbf{r}}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k) \cdot \mathbf{r}_t}{\sum_{t=1}^{T} \gamma_t(j,k)} \tag{6.63a}$$

where $\mathbf{r}_t = [r_t(0), r_t(1), \ldots, r_t(p)]'$ is the autocorrelation vector as defined by Eq. (6.57d) for the $t^{\text{th}}$ frame, and $\gamma_t(j,k)$ is defined as the probability of being in state $j$ at time $t$ and using mixture component $k$, i.e.,

$$\gamma_t(j,k) = \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}\right]\left[\frac{c_{jk} b_{jk}(\mathbf{o}_t)}{\sum_{k=1}^{M} c_{jk} b_{jk}(\mathbf{o}_t)}\right]. \tag{6.63b}$$

It can be seen that $\bar{\mathbf{r}}_{jk}$ is a weighted sum (by probability of occurrence) of the normalized autocorrelations of the frames in the observation sequence. From $\bar{\mathbf{r}}_{jk}$, one can solve a set of normal equations to obtain the corresponding autoregressive coefficient vector $\bar{\mathbf{a}}_{jk}$, for the $k$th mixture of state $j$. The new autocorrection vectors of the autoregression coefficients as needed in the density function can then be calculated using Eq. (6.57c), thereby closing the reestimation loop.

**Exercise 6.4**

The probability density function (pdf) [22, 23] of Eq. (6.56) is defined by parameters $\sigma_\epsilon^2$ and $\mathbf{a}$. Given a data observation vector $\mathbf{o} = (x_0, x_1, \ldots, x_{K-1})$, determine the maximum likelihood estimate of $\sigma_\epsilon^2$ and $\mathbf{a}$ that best characterizes the observed $\mathbf{o}$.

**Solution 6.4**

We write the likelihood function of $\mathbf{o}$ as a function of $\sigma_\epsilon^2$ and $\mathbf{a}$ as

$$f(\mathbf{o}|\sigma_\epsilon^2, \mathbf{a}) = (2\pi\sigma_\epsilon^2)^{-K/2} \exp\left\{-\frac{1}{2\sigma_\epsilon^2}\delta(\mathbf{o}, \mathbf{a})\right\}$$

and the log likelihood function as

$$\log f(\mathbf{o}|\sigma_\epsilon^2, \mathbf{a}) = -\frac{K}{2}\log(2\pi\sigma_\epsilon^2) - \frac{\delta(\mathbf{o}, \mathbf{a})}{2\sigma_\epsilon^2}.$$



Therefore, the maximum likelihood (ML) estimate is

$$\begin{aligned}(\mathbf{a})_{\text{ML}} &= \arg\max_{\mathbf{a}} \log f(\mathbf{o}|\sigma_\epsilon^2, \mathbf{a}) = \arg\min_{\mathbf{a}} \delta(\mathbf{o}, \mathbf{a}) \\ &= \arg\min_{\mathbf{a}} (\mathbf{a}'\mathbf{R}\mathbf{a})\end{aligned}$$

where $\mathbf{R} = [r_{ij}]$ with $r_{ij} = r(|i-j|)$ as defined by Eq. (6.57d). This establishes the relationship between maximum likelihood estimation and the classical method of autocorrelation matching (also called the method of minimization of prediction residuals) for LPC analysis (see Eq. (4.8)–(4.10)). Furthermore, it is easy to show that

$$(\sigma_\epsilon^2)_{\text{ML}} = \min_{\mathbf{a}} \delta(\mathbf{o}, \mathbf{a})/K$$

which leads to

$$\max_{\sigma_\epsilon^2, \mathbf{a}} \log f(\mathbf{o}|\sigma_\epsilon^2, \mathbf{a}) = -\frac{K}{2}\log\left[2\pi(\sigma_\epsilon^2)_{\text{ML}}\right] - \frac{K}{2}.$$

**Exercise 6.5**

The pdf of Eq. (6.56) is related to the Itakura-Saito distortion measure of Eq. (4.45). Establish this relationship.

**Solution 6.5**

Consider the following likelihood difference

$$\begin{aligned}L_d &= \left[\max_{\sigma_\epsilon^2, \mathbf{a}} \log f(\mathbf{o}|\sigma_\epsilon^2, \mathbf{a})\right] - \log f(\mathbf{o}|\sigma_\epsilon^2, \mathbf{a}) \\ &= -\frac{K}{2}\log\left[2\pi(\sigma_\epsilon^2)_{\text{ML}}\right] - \frac{K}{2} + \frac{K}{2}\log(2\pi\sigma_\epsilon^2) + \frac{\delta(\mathbf{o}, \mathbf{a})}{2\sigma_\epsilon^2} \\ &= \frac{K}{2}\left[\frac{1}{K\sigma_\epsilon^2}\delta(\mathbf{o}, \mathbf{a}) + \log\sigma_\epsilon^2 - \log(\sigma_\epsilon^2)_{\text{ML}} - 1\right].\end{aligned}$$

For distortion measures, we denote the all-pole (power) spectra by $\sigma_{\epsilon_0}^2/|A_0(e^{j\omega})|^2$ and $\sigma_\epsilon^2/|A(e^{j\omega})|^2$, corresponding to parameter sets $\{(\sigma_\epsilon^2)_{\text{ML}}, (\mathbf{a})_{\text{ML}}\}$ and $\{\sigma_\epsilon^2, \mathbf{a}\}$, respectively. ($\sigma_{f_0}^2 = (\sigma_\epsilon^2)_{\text{ML}}$.) Then, the bracketed term in the log likelihood difference, $L_d$, is simply $d_{\text{IS}}(\sigma_{f_0}^2/|A_0|^2, \sigma_\epsilon^2/|A|^2)$ according to Eq. (4.45) because

$$\sigma_{f_0}^2 \int_{-\pi}^{\pi} \frac{|A(e^{j\omega})|^2}{|A_0(e^{j\omega})|^2}\frac{d\omega}{2\pi} = \frac{1}{K}\delta(\mathbf{o}, \mathbf{a})$$

due to autocorrelation matching and Eq. (6.57a). (Note that we have defined $\sigma_\epsilon^2$ (and $\sigma_{\epsilon_0}^2$) as the sample variance. The factor $K$ would disappear from the above equation if we use the total frame prediction residual variance instead of the sample variance.) Therefore,

$$\begin{aligned}f(\mathbf{o}|\sigma_\epsilon^2, \mathbf{a}) &= (2\pi\sigma_\epsilon^2)^{-K/2} \exp\left\{-\frac{K}{2}\left[d_{\text{IS}}\left(\frac{\sigma_{\epsilon_0}^2}{|A_0|^2}, \frac{\sigma_\epsilon^2}{|A|^2}\right) + \log\sigma_{f_0}^2 - \log\sigma_\epsilon^2 + 1\right]\right\} \\ &= G_1(\sigma_{f_0}^2, \sigma_\epsilon^2) \exp\left\{-\frac{K}{2} d_{\text{IS}}\left(\frac{\sigma_{f_0}^2}{|A_0|^2}, \frac{\sigma_\epsilon^2}{|A|^2}\right)\right\}\end{aligned}$$

where $G_1(\sigma_{fo}^2, \sigma_e^2)$ encompasses only the gain terms.

**Exercise 6.6**

The pdf of Eq. (6.60) is related to the likelihood ratio distortion measure of Eq. (4.53). Establish this relationship.

**Solution 6.6**

Similar to the case of Eq. (6.56),

$$\frac{1}{K}\delta(\hat{\mathbf{o}}, \mathbf{a}) = \int_{-\pi}^{\pi} \frac{|A(e^{j\omega})|^2}{|A_o(e^{j\omega})|^2} \frac{d\omega}{2\pi}$$
$$= d_{LR}\left(\frac{1}{|A_o|^2}, \frac{1}{|A|^2}\right) + 1.$$

Therefore,

$$f(\hat{\mathbf{o}}|\mathbf{a}) = (2\pi)^{-K/2} \exp\left\{-\frac{K}{2}\left[d_{LR}\left(\frac{1}{|A_o|^2}, \frac{1}{|A|^2}\right) + 1\right]\right\}$$
$$= G_2 \exp\left\{-\frac{K}{2} d_{LR}\left(\frac{1}{|A_o|^2}, \frac{1}{|A|^2}\right)\right\}.$$

Note that when the pdf is expressed in terms of the distortion measures ($d_{IS}$ or $d_{LR}$), the exponential term includes a factor $K$ that represents the data dimension. In practice, this factor $K$ is replaced by an effective frame length $\hat{K}$, which is the net shift of consecutive data frames. Thus, if consecutive data frames (vectors) have 2/3 overlap, then an effective frame length $\hat{K} = K/3$ is appropriate, so that the rate of characteristic change in terms of the spectral parameters a is kept at the original waveform sampling rate.

## 6.8 VARIANTS ON HMM STRUCTURES—NULL TRANSITIONS AND TIED STATES

Throughout this chapter we have considered HMMs in which the observations were associated with states of the model. It is also possible to consider models in which the observations are associated with the arcs of the model. This type of HMM has been used extensively in the IBM continuous speech recognizer [13]. It has been found useful, for this type of model, to allow transitions that produce no output; that is, jumps from one state to another that produce no observation [13]. Such transitions are called null transitions and are designated by a dashed line, with the symbol $\phi$ used to denote the null output.

Figure 6.10 illustrates three examples (from speech-processing tasks) where null arcs have been successfully utilized. The example of part (a) corresponds to an HMM (a left-right model) with a large number of states in which it is possible to omit transitions between any pair of states. Hence it is possible to generate observation sequences with as few as



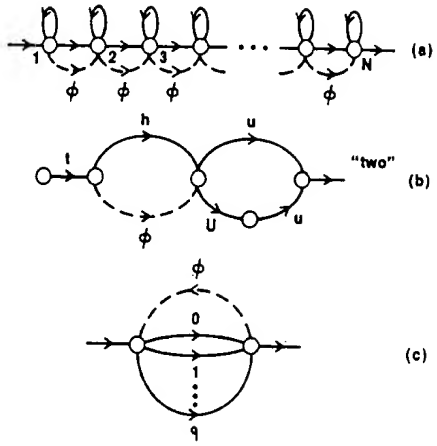



**Figure 6.10** Examples of networks incorporating null transitions. (a) Left-right model. (b) Finite state network. (c) Grammar network.

one observation and still account for a path that begins in state one and ends in state $N$.

The example of Figure 6.10(b) is a finite-state network (FSN) representation of a word in terms of linguistic unit models (i.e., the sound on each arc is itself an HMM). For this model the null transition gives a compact and efficient way of describing alternative word pronunciations (i.e., symbol deletions).

Finally the FSN of Figure 6.10(c) shows how the ability to insert a null transition into a grammar network allows a relatively simple network to generate arbitrarily long word (digit) sequences. In the example shown in Figure 6.10(c), the null transition allows the network to generate arbitrary sequences of digits of arbitrary length by returning to the initial state after each individual digit is produced.

Another interesting variation in the HMM structure is the concept of parameter tying [13]. Basically, the idea is to set up an equivalence relation between HMM parameters in different states. In this manner the number of independent parameters in the model is reduced and the parameter estimation becomes somewhat simpler and in some cases more reliable. Parameter tieing is used when the observation density, for example, is known to be the same in two or more states. Such cases occur often in characterizing speech sounds. The technique is especially appropriate when there is insufficient training data to estimate, reliably, a large number of model parameters. For cases such as these, it is appropriate to tie model parameters so as to reduce the number of parameters (i.e., size of the model), thereby making the parameter estimation problem somewhat simpler. We will discuss this method later in this chapter.

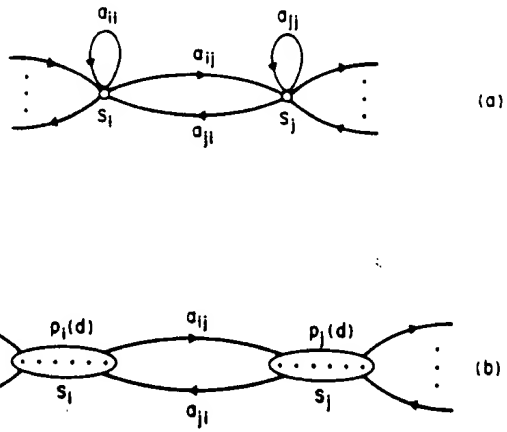



**Figure 6.11** Illustration of general interstate connections of (a) a normal HMM with exponential state duration density, and (b) a variable duration HMM with specified state densities and no self-transitions from a state back to itself.

## 6.9 INCLUSION OF EXPLICIT STATE DURATION DENSITY IN HMMS

Earlier we showed via Eq. (6.5) that the inherent duration probability density $p_i(d)$ associated with state $i$, with self-transition coefficient $a_{ii}$, was of the form

$$\begin{aligned} p_i(d) &= (a_{ii})^{d-1}(1 - a_{ii}) \\ &= \text{probability of } d \text{ consecutive observations in state } i. \end{aligned} \tag{6.64}$$

For most physical signals, this exponential state duration density is inappropriate. Instead we would prefer to explicitly model duration density in some analytic form. (An extensive treatment of state duration modeling can be found in the work of Ferguson of IDA [14], which is the basis of the material presented here. Other valuable references include [24] and [25].) Figure 6.11 illustrates, for a pair of model states $i$ and $j$, the differences between HMMs without and with explicit duration density. In part (a) the states have exponential duration densities based on self-transition coefficients $a_{ii}$ and $a_{jj}$, respectively. In part (b), the self-transition coefficients are set to zero, and an explicit duration density is specified. For this case, a transition is made only after the appropriate number of observations have occurred in the state (as specified by the duration density). Such a model is called a semi-Markov model.

Based on the simple model of Figure 6.11(b), the sequence of events of the variable duration HMM is as follows:

1. An initial state, $q_1 = i$, is chosen according to the initial state distribution $\pi_i$.



2. A duration $d_1$ is chosen according to the state duration density $p_{q_1}(d_1)$. (For expedience and ease of implementation, the duration density $p_q(d)$ is truncated at a maximum duration value $D$.)
3. Observations $\mathbf{o}_1\,\mathbf{o}_2 \ldots \mathbf{o}_{d_1}$ are chosen according to the joint observation density, $b_{q_1}(\mathbf{o}_1\,\mathbf{o}_2 \ldots \mathbf{o}_{d_1})$. Generally we assume that in each state observations are independent so that $b_{q_1}(\mathbf{o}_1\,\mathbf{o}_2 \ldots \mathbf{o}_{d_1}) = \prod_{t=1}^{d_1} b_{q_1}(\mathbf{o}_t)$.
4. The next state, $q_2 = j$, is chosen according to the state transition probabilities, $a_{q_1 q_2}$, with the constraint that $a_{q_1 q_1} = 0$, i.e., no transition back to the same state can occur. (Clearly this is a requirement, because we assume that, in state $q_1$, exactly $d_1$ observations occur.)

A little thought should convince the reader that the variable duration HMM can be made equivalent to the standard HMM by setting $p_i(d)$ to be the exponential density of (6.64).

Using the above formulation, we must make several changes to the formulas of Section 6.4.3 to allow calculation of $P(\mathbf{O}|\lambda)$ and for reestimation of all model parameters. In particular we assume that the first state begins at $t = 1$ and the last state ends at $t = T$. We then define the forward variable $\alpha_t(i)$ as

$$\alpha_t(i) = P(\mathbf{o}_1\,\mathbf{o}_2 \ldots \mathbf{o}_t, \text{ the stay in state } i \text{ ends at } t|\lambda). \tag{6.65}$$

We assume that a total of $r$ states have been visited during the first $t$ observations, and we denote the states as $q_1, q_2, \ldots, q_r$ with durations associated with each state of $d_1, d_2, \ldots, d_r$. Thus the constraints of Eq. (6.65) are

$$q_r = i \tag{6.66a}$$

$$\sum_{s=1}^{r} d_s = t. \tag{6.66b}$$

Eq. (6.65) can then be written as

$$\begin{aligned} \alpha_t(i) = \sum_q \sum_d \pi_{q_1} \cdot p_{q_1}(d_1) \cdot P(\mathbf{o}_1\,\mathbf{o}_2 \ldots \mathbf{o}_{d_1}|q_1) \\ \cdot a_{q_1 q_2} p_{q_2}(d_2) P(\mathbf{o}_{d_1+1} \ldots \mathbf{o}_{d_1+d_2}|q_2) \ldots \\ \cdot a_{q_{r-1} q_r} p_{q_r}(d_r) P(\mathbf{o}_{d_1+d_2+\cdots+d_{r-1}+1} \ldots \mathbf{o}_t|q_r) \end{aligned} \tag{6.67}$$

where the sum is over all states $q$ and all possible state durations $d$. By induction we can write $\alpha_t(j)$ as

$$\alpha_t(j) = \sum_{i=1}^{N} \sum_{d=1}^{D} \alpha_{t-d}(i) a_{ij} p_j(d) \prod_{s=t-d+1}^{t} b_j(\mathbf{o}_s) \tag{6.68}$$

where $D$ is the maximum duration within any state. To initialize the computation of $\alpha_t(j)$ we use

$$\alpha_1(i) = \pi_i p_i(1) \cdot b_i(\mathbf{o}_1) \tag{6.69a}$$

$$\alpha_2(i) = \pi_i p_i(2) \prod_{s=1}^{2} b_i(\mathbf{o}_s) + \sum_{\substack{j=1 \\ j \neq i}}^{N} \alpha_1(j) a_{ji} p_i(1) b_i(\mathbf{o}_2) \tag{6.69b}$$

$$\alpha_3(i) = \pi_i p_i(3) \prod_{s=1}^{3} b_i(\mathbf{o}_s) + \sum_{d=1}^{2} \sum_{\substack{j=1 \\ j \neq i}}^{N} \alpha_{3-d}(j) a_{ji} p_i(d) \tag{6.69c}$$

$$\cdot \prod_{s=4-d}^{3} b_i(\mathbf{o}_s) \tag{6.69d}$$

and so on, until $\alpha_D(i)$ is computed; then Eq. (6.68) can be used for all $t > D$. It should be clear that the desired probability of O given the model $\lambda$ can be written in terms of the $\alpha$s as

$$P(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{6.70}$$

as was previously used for ordinary HMMs.

To give reestimation formulas for all the variables of the variable duration HMM, we must define three more forward-backward variables, namely

$$\alpha_t^*(i) = P(\mathbf{o}_1 \mathbf{o}_2 \ldots \mathbf{o}_t, \text{ stay in state } i \text{ starts at } t+1|\lambda) \tag{6.71}$$

$$\beta_t(i) = P(\mathbf{o}_{t+1} \ldots \mathbf{o}_T| \text{ stay in state } i \text{ ends at } t, \lambda) \tag{6.72}$$

$$\beta_t^*(i) = P(\mathbf{o}_{t+1} \ldots \mathbf{o}_T| \text{ stay in state } i \text{ starts at } t+1, \lambda). \tag{6.73}$$

The relationships between $\alpha, \alpha^*, \beta$, and $\beta^*$ are as follows:

$$\alpha_t^*(j) = \sum_{i=1}^{N} \alpha_t(i) a_{ij} \tag{6.74}$$

$$\alpha_t(i) = \sum_{d=1}^{D} \alpha_{t-d}^*(i) p_i(d) \prod_{s=t-d+1}^{t} b_i(\mathbf{o}_s) \tag{6.75}$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} \beta_t^*(j) \tag{6.76}$$

$$\beta_t^*(i) = \sum_{d=1}^{D} \beta_{t+d}(i) p_i(d) \prod_{s=t+1}^{t+d} b_i(\mathbf{o}_s). \tag{6.77}$$

Based on the above relationships and definitions, the reestimation formulas for the variable duration HMM, with discrete observations, are

$$\bar{\pi}_i = \frac{\pi_i \beta_0^*(i)}{P(\mathbf{O}|\lambda)} \tag{6.78}$$



$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T} \alpha_t(i) a_{ij} \beta_t^*(j)}{\sum_{j=1}^{N} \sum_{t=1}^{T} \alpha_t(i) a_{ij} \beta_t^*(j)} \tag{6.79}$$

$$\bar{b}_i(k) = \frac{\sum_{\substack{t=1 \\ \text{s.t. } \mathbf{o}_t = \mathbf{v}_k}}^{T} \left[ \sum_{\tau < t} \alpha_\tau^*(i) \cdot \beta_\tau^*(i) - \sum_{\tau < t} \alpha_\tau(i) \beta_\tau(i) \right]}{\sum_{k=1}^{M} \sum_{\substack{t=1 \\ \text{s.t. } \mathbf{o}_t = \mathbf{v}_k}}^{T} \left[ \sum_{\tau < t} \alpha_\tau^*(i) \cdot \beta_\tau^*(i) - \sum_{\tau < t} \alpha_\tau(i) \beta_\tau(i) \right]} \tag{6.80}$$

$$\bar{p}_i(d) = \frac{\sum_{t=1}^{T} \alpha_t^*(i) p_i(d) \beta_{t+d}(i) \prod_{s=t+1}^{t+d} b_i(\mathbf{o}_s)}{\sum_{d=1}^{D} \sum_{t=1}^{T} \alpha_t^*(i) p_i(d) \beta_{t+d}(i) \prod_{s=t+1}^{t+d} b_i(\mathbf{o}_s)}. \tag{6.81}$$

The interpretation of the reestimation formulas is the following: The formula for $\bar{\pi}_i$ is the probability that state $i$ was the first state, given O. The formula for $\bar{a}_{ij}$ is almost the same as for the usual HMM, except it uses the condition that the alpha terms in which a state ends at $t$, join with the beta terms in which a new state begins at $t+1$. The formula for $\bar{b}_i(k)$ is the expected number of times that observation $\mathbf{o}_t = \mathbf{v}_k$ occurred in state $i$, normalized by the expected number of times that any observation occurred in state $i$. Finally, the reestimation formula for $\bar{p}_i(d)$ is the ratio of the expected number of times state $i$ occurred with any duration.

The importance of incorporating state duration densities is reflected in the observation that, for some problems, the quality of the modeling is significantly improved when explicit state duration densities are used. However, there are drawbacks to the use of the variable duration model discussed in this section. One is the greatly increased computational load associated with using variable durations. It can be seen from the definition and initialization conditions on the forward variable $\alpha_i(i)$, from Eqs. (6.68)–(6.69), that about $D$ times the storage and $D^2/2$ times the computation is required. For $D$ on the order of 25 (as is reasonable for many speech-processing problems), computation is increased by a factor of 300. Another problem with the variable duration models is the large number of parameters ($D$), associated with each state, that must be estimated, in addition to the usual HMM parameters. Furthermore, for a fixed number of observations $T$, in the training set, there are, on average, fewer state transitions and much less data to estimate $p_i(d)$ than would be used in a standard HMM. Thus the reestimation problem is more difficult for variable duration HMMs than for the standard HMM.

One proposal to alleviate some of these problems is to use a parametric state duration density instead of the nonparametric $p_i(d)$ used above [23–24]. In particular, proposals

include the Gaussian family with

$$p_i(d) = \mathcal{N}(d, \mu_i, \sigma_i^2) \tag{6.82}$$

with parameters $\mu_i$ and $\sigma_i^2$, or the Gamma family with

$$p_i(d) = \frac{\eta_i^{\nu_i} d^{\nu_i - 1} e^{-\eta_i d}}{\Gamma(\nu_i)} \tag{6.83}$$

with parameters $\nu_i$ and $\eta_i$ and with mean $\nu_i \eta_i^{-1}$ and variance $\nu_i \eta_i^{-2}$. Reestimation formulas for $\eta_i$ and $\nu_i$ have been derived and used with good results. Another possibility, which has been used with good success, is to assume a uniform duration distribution over an appropriate range of durations and use a path-constrained Viterbi decoding procedure.

## 6.10 OPTIMIZATION CRITERION—ML, MMI, AND MDI

The standard ML design criterion is to use a training sequence of observations $\mathbf{O}$ to derive the set of model parameters $\lambda$, yielding

$$\lambda_{\text{ML}} = \arg\max_{\lambda} P(\mathbf{O}|\lambda). \tag{6.84}$$

Any of the reestimation algorithms discussed previously provides a solution to this optimization problem.

The need to consider alternative design criteria, however, comes from several concerns ([26–28]). The basic philosophy in statistical modeling methods, such as HMM, is that the signal or observation sequence can be well modeled if the parameters of the model are carefully and correctly chosen. The problem with this philosophy is that the assumed model—HMM in the present case—is sometimes inadequate to model the observed signal so that no matter how carefully the parameters are chosen, the modeling accuracy is limited. Often, this situation is described as a "model mismatch." The first alternative optimization criterion we discuss here is one that tries to overcome the problem of model mismatch in order to achieve a more accurate modeling of the observation signal.

The observed signal $\mathbf{O} = (\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_T)$ is associated with a sequence of constraints $\mathcal{R} = (\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_T)$. For example, $\mathbf{R}_t$ may be the autocorrelation matrix that characterizes the observation $\mathbf{o}_t$. Then, obviously, $\mathbf{O}$ is only one of possibly uncountably many observation sequences that satisfy the constraint sequence $\mathcal{R}$. Furthermore, in terms of the probability distributions of the observation sequences, there exists a set of such distributions that would also satisfy $\mathcal{R}$. We denote this set $\Omega(\mathcal{R})$. The minimum discrimination information (MDI) is a measure of closeness between two probability measures (one of which bears the HMM form here) under the given constraint $\mathcal{R}$ and is defined by

$$\nu(\mathcal{R}, P_\lambda) \triangleq \inf_{Q \in \Omega(\mathcal{R})} I(Q : P_\lambda) \tag{6.85}$$

where

$$I(Q : P_\lambda) = \int q(\mathbf{O}) \log \frac{q(\mathbf{O})}{p(\mathbf{O}|\lambda)} d\mathbf{O} \tag{6.86}$$



is the discrimination information between distributions $Q$ and $P_\lambda$ [27,28]. (The functions $q(\cdot)$ and $p(\cdot|\lambda)$ are the probability density functions corresponding to $Q$ and $P_\lambda$ respectively.) The discrimination information is calculated based on the given training set of observations.

The MDI criterion tries to choose a model parameter set $\lambda$ such that $\nu(\mathcal{R}, P_\lambda)$ is minimized. An interpretation of MDI is that the model parameter set $\lambda$ is chosen so that the model $p(\mathbf{O}|\lambda)$ is as close as it can be to a member of the set $\Omega(\mathcal{R})$. Since the closeness is always measured in terms of the discrimination information evaluated on the given observation, the intrinsic characteristics of the training sequences would then have substantial influence on the parameter selection. By emphasizing the measure discrimination, the model estimation is no longer solely dictated by the assumed model form. The MDI optimization problem is, however, not as straightforward as the ML optimization problem and no simple robust implementation of the procedure is known.

Another concern about the HMM optimization criterion arises when we attempt to use it to solve a class of speech-recognition problems. Consider recognition of a vocabulary of $V$ words, each of which is represented by an HMM, with parameter set $\lambda_v$, $v = 1, 2, \ldots, V$. We assume $P(v)$ to be the a priori probability for word $v$, $v = 1, 2, \ldots, V$. The set of HMMs $\Lambda = \{\lambda_v\}$ together with the a priori probabilities thus defines a probability measure for an arbitrary observation sequence $\mathbf{O}$

$$P_\Lambda(\mathbf{O}) = \sum_{v=1}^{V} P(\mathbf{O}|\lambda_v, v) P(v). \tag{6.87}$$

(The notation $P(\mathbf{O}|\lambda_v, v)$ indicates that it is a probability conditioned on the word $v$. We include the model parameter $\lambda_v$, sometimes, because of the necessity of treating $\lambda_v$ as random variables for estimation purposes. Obviously, when $\lambda_v$ is fixed, $P(\mathbf{O}|\lambda_v, v)$ is the conditional probability, parameterized by $\lambda_v$.) To train these models (i.e., to estimate the optimum parameters of the associated models), utterances of known (labeled) words are used. We denote the labeled training sequence by $\mathbf{O}^v$ where superscript $v$ reflects the fact that $\mathbf{O}^v$ is a rendition of word $v$. The standard ML criterion of Eq. (6.84) is to use $\mathbf{O}^v$ to estimate model parameters $\lambda_v$, yielding

$$(\lambda_v)_{\text{ML}} = \arg\min_{\lambda} P(\mathbf{O}^v|\lambda).$$

Each model is estimated *separately* using the correspondingly labeled training observation sequence(s). The resultant models, however, need not be the optimal solution for minimizing the probability of recognition error.

An alternative design criterion that aims at maximizing the "discrimination" of each model (i.e., the ability to distinguish between observation sequences generated by the correct word model and those generated by alternative models) is the maximum mutual information (MMI) criterion [26]. The mutual information between an observation sequence $\mathbf{O}^v$ and the word $v$, parameterized by $\Lambda = \{\lambda_v\}$, $v = 1, 2, \ldots, V$, is

$$I_\Lambda(\mathbf{O}^v, v) = \log \frac{P(\mathbf{O}^v, v|\Lambda)}{P_\Lambda(\mathbf{O}^v) P(v)}. \tag{6.88}$$

Since

$$P(\mathbf{O}^v, v|\Lambda)/P(v) = P(\mathbf{O}^v|\lambda_v),$$

$$I_\Lambda(\mathbf{O}^v, v) = \log\ P(\mathbf{O}^v|\lambda_v) - \log \sum_{w=1}^{V} P(\mathbf{O}^v|\lambda_w, w)P(w). \tag{6.89}$$

The MMI criterion is to find the entire model set $\Lambda$ such that the mutual information is maximized,

$$(\Lambda)_{\text{MMI}} = \max_{\Lambda}\left\{\sum_{v=1}^{V} I_\Lambda(\mathbf{O}^v, v)\right\}. \tag{6.90}$$

The MMI criterion is obviously different from the ML criterion. Both are minimum cross-entropy approaches. In the ML approach, an HMM for the distribution of the *data given the word* is matched to the empirical distribution. In the MMI approach, a model for the distribution of the *word given the data* is matched to the empirical distribution. This explains the merit of the MMI approach. The optimization procedure for the MMI approach involves the entire model parameter set $\Lambda$ even if only one labeled training sequence $\mathbf{O}^v$ is used. The ML criterion addresses the likelihood $P(\mathbf{O}^v|\lambda_v)$ alone, while the MMI criterion compares the likelihood $P(\mathbf{O}^v|\lambda_v)$ against the "probability background" $P_\Lambda(\mathbf{O}^v)$ and attempts to maximize the difference. However, $(\Lambda)_{\text{MMI}}$ is not as straightforward to obtain as $(\Lambda)_{\text{ML}}$. One often has to use general optimization procedures like the descent algorithms to solve Eq. (6.90). Such optimization procedures often lead to numerical problems in implementation.

## 6.11 COMPARISONS OF HMMS

An interesting question associated with HMMs is the following: Given two HMMs, $\lambda_1$ and $\lambda_2$, what is a reasonable measure of the similarity of the two models ([29])? Consider the case of two models

$$\lambda_1 = (A_1, B_1, \pi_1) \qquad \lambda_2 = (A_2, B_2, \pi_2)$$

with

$$A_1 = \begin{bmatrix} p & 1-p \\ 1-p & p \end{bmatrix} \quad B_1 = \begin{bmatrix} q & 1-q \\ 1-q & q \end{bmatrix} \quad \pi_1 = [1/2\ \ 1/2]$$

and

$$A_2 = \begin{bmatrix} r & 1-r \\ 1-r & r \end{bmatrix} \quad B_2 = \begin{bmatrix} s & 1-s \\ 1-s & s \end{bmatrix} \quad \pi_2 = [1/2\ \ 1/2].$$

For $\lambda_1$ to be equivalent to $\lambda_2$, in the sense of having the same statistical properties for the observation symbols, i.e., $E[\mathbf{o}_t = \mathbf{v}_k|\lambda_1] = E[\mathbf{o}_t = \mathbf{v}_k|\lambda_2]$, for all $\mathbf{v}_k$, we require

$$pq + (1-p)(1-q) = rs + (1-r)(1-s)$$



or, by solving for $s$, we get

$$s = \frac{p + q - 2pq}{1 - 2r}.$$

By choosing (arbitrarily) $p = 0.6, q = 0.7, r = 0.2$, we get $s = 13/30 \approx 0.433$. Thus, even when the two models, $\lambda_1$ and $\lambda_2$, look ostensibly very different (i.e., $A_1$ is very different from $A_2$ and $B_1$ is very different from $B_2$), statistical equivalence of the models can occur.

We can generalize [29] the concept of model distance (dissimilarity) by defining a distance measure $D(\lambda_1, \lambda_2)$, between two Markov models, $\lambda_1$ and $\lambda_2$, as

$$D(\lambda_1, \lambda_2) = \frac{1}{T}\left[\log\ P(\mathbf{O}^{(2)}|\lambda_1) - \log\ P(\mathbf{O}^{(2)}|\lambda_2)\right] \tag{6.91}$$

where $\mathbf{O}^{(2)} = (\mathbf{o}_1\, \mathbf{o}_2\, \mathbf{o}_3 \ldots \mathbf{o}_T)$ is a sequence of observations generated by model $\lambda_2$. Basically, Eq. (6.91) is a measure of how well model $\lambda_1$ matches observations generated by model $\lambda_2$, relative to how well model $\lambda_2$ matches observations generated by itself. Several interpretations of Eq. (6.91) exist in terms of cross-entropy, or divergence, or discrimination information [29].

One of the problems with the distance measure of Eq. (6.91) is that it is nonsymmetric. Hence a natural expression of this measure is the symmetrized version, namely

$$D_s(\lambda_1, \lambda_2) = \frac{D(\lambda_1\lambda_2) + D(\lambda_2, \lambda_1)}{2}. \tag{6.92}$$

## 6.12 IMPLEMENTATION ISSUES FOR HMMS

The discussion in the previous sections has dealt primarily with the theory of HMMs and several variations on the form of the model. In this section we deal with several practical implementation issues, including scaling, multiple observation sequences, initial parameter estimates, missing data, and choice of model size and type. For some of these implementation issues we can prescribe exact analytical solutions; for other issues we can provide only some seat-of-the-pants experience gained from working with HMMs.

### 6.12.1 Scaling

To understand why scaling ([18,23]) is required for implementing the reestimation procedure of HMMs, consider the definition of $\alpha_t(i)$ of Eq. (6.18). It can be seen that $\alpha_t(i)$ consists of the sum of a large number of terms, each of the form

$$\left(\prod_{s=1}^{t-1} a_{q_s q_{s+1}} \prod_{s=1}^{t} b_{q_s}(\mathbf{o}_s)\right)$$

with $q_t = i$ and $b$ is a discrete probability as defined by Eq. (6.8). Since each $a$ and $b$ term is less than 1 (generally significantly less than 1), it can be seen that as $t$ starts to get big (e.g., 10 or more), each term of $\alpha_t(i)$ starts to head exponentially to zero. For sufficiently large $t$ (e.g., 100 or more) the dynamic range of the $\alpha_t(i)$ computation will exceed the precision

range of essentially any machine (even in double precision). Hence the only reasonable way to perform the computation is to incorporate a scaling procedure.

The basic scaling procedure multiplies $\alpha_t(i)$ by a scaling coefficient that is independent of $i$ (i.e., it depends only on $t$), with the goal of keeping the scaled $\alpha_t(i)$ within the dynamic range of the computer for $1 \le t \le T$. A similar scaling is done to the $\beta_t(i)$ coefficients (since these also tend to zero exponentially fast) and then, at the end of the computation, the scaling coefficients are canceled out exactly.

To understand this scaling procedure better, consider the reestimation formula for the state-transition coefficients $a_{ij}$. If we write the reestimation formula (Eq. (6.40b)) directly in terms of the forward and backward variables, we get

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j)}. \tag{6.93}$$

Consider the computation of $\alpha_t(i)$. We use the notation $\alpha_t(i)$ to denote the unscaled $\alpha$s, $\hat{\alpha}_t(i)$ to denote the scaled (and iterated) $\alpha$s, and $\widehat{\alpha}_t(i)$ to denote the local version of $\alpha$ before scaling. Initially, for $t = 1$, we compute $\alpha_1(i)$ according to Eq. (6.19) and set $\widehat{\alpha}_1(i) = \alpha_1(i)$, with $c_1 = \frac{1}{\sum_{i=1}^{N} \alpha_1(i)}$ and $\hat{\alpha}_1(i) = c_1 \alpha_1(i)$. For each $t$, $2 \le t \le T$, we first compute $\widehat{\alpha}_t(i)$ according to the induction formula (Eq. (6.20)), in terms of the previously scaled $\hat{\alpha}_t(i)$; that is,

$$\widehat{\alpha}_t(i) = \sum_{j=1}^{N} \hat{\alpha}_{t-1}(j) a_{ji} b_i(\mathbf{o}_t). \tag{6.94a}$$

We determine the scaling coefficient $c_t$ as

$$c_t = \frac{1}{\sum_{i=1}^{N} \widehat{\alpha}_t(i)} \tag{6.94b}$$

giving

$$\hat{\alpha}_t(i) = c_t \widehat{\alpha}_t(i) \tag{6.94c}$$

From Eq. (6.94a–c) we can write the scaled $\hat{\alpha}_t(i)$ as $c_t \widehat{\alpha}_t(i)$ or

$$\hat{\alpha}_t(i) = \frac{\sum_{j=1}^{N} \hat{\alpha}_{t-1}(j) a_{ji} b_i(\mathbf{o}_t)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \hat{\alpha}_{t-1}(j) a_{ji} b_i(\mathbf{o}_t)}. \tag{6.95}$$



By induction we can write $\hat{\alpha}_{t-1}(j)$ as

$$\hat{\alpha}_{t-1}(j) = \left( \prod_{\tau=1}^{t-1} c_\tau \right) \alpha_{t-1}(j). \tag{6.96a}$$

Thus we can write $\hat{\alpha}_t(i)$ as

$$\hat{\alpha}_t(i) = \frac{\sum_{j=1}^{N} \alpha_{t-1}(j) \left( \prod_{\tau=1}^{t-1} c_\tau \right) a_{ji} b_i(\mathbf{o}_t)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_{t-1}(j) \left( \prod_{\tau=1}^{t-1} c_\tau \right) a_{ji} b_i(\mathbf{o}_t)} = \frac{\alpha_t(i)}{\sum_{i=1}^{N} \alpha_t(i)} \tag{6.96b}$$

that is, each $\alpha_t(i)$ is effectively scaled by the sum over all states of $\alpha_t(i)$.

Next we compute the $\beta_t(i)$ terms from the backward recursion. The only difference here is that we use the *same* scale factors for each time $t$ for the betas as was used for the alphas. Hence the scaled $\beta$s are of the form

$$\hat{\beta}_t(i) = c_t \beta_t(i). \tag{6.97}$$

Because each scale factor effectively restores the magnitude of the $\alpha$ terms to 1, and because the magnitudes of the $\alpha$ and $\beta$ terms are comparable, using the same scaling factors on the $\beta$s as was used on the $\alpha$s is an effective way to keep the computation within reasonable bounds. Furthermore, in terms of the scaled variables, we see that the reestimation Eq. (6.93) becomes

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \hat{\alpha}_t(i) a_{ij} b_j(\mathbf{o}_{t+1}) \hat{\beta}_{t+1}(j)}{\sum_{t=1}^{T-1} \sum_{j=1}^{N} \hat{\alpha}_t(i) a_{ij} b_j(\mathbf{o}_{t+1}) \hat{\beta}_{t+1}(j)} \tag{6.98}$$

but each $\hat{\alpha}_t(i)$ can be written as

$$\hat{\alpha}_t(i) = \left[ \prod_{s=1}^{t} c_s \right] \alpha_t(i) = C_t \alpha_t(i) \tag{6.99}$$

and each $\hat{\beta}_{t+1}(j)$ can be written as

$$\hat{\beta}_{t+1}(j) = \left[ \prod_{s=t+1}^{T} c_s \right] \beta_{t+1}(j) = D_{t+1} \beta_{t+1}(j). \tag{6.100}$$

Thus Eq. (6.98) can be written as

$$\overline{a_{ij}} = \frac{\sum_{t=1}^{T-1} C_t \alpha_t(i) a_{ij} b_j(\mathbf{o}_{t+1}) D_{t+1} \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \sum_{j=1}^{N} C_t \alpha_t(i) a_{ij} b_j(\mathbf{o}_{t+1}) D_{t+1}\, \beta_{t+1}(j)}. \tag{6.101}$$

Finally the term $C_t D_{t+1}$ can be seen to be of the form

$$C_t D_{t+1} = \prod_{s=1}^{t} c_s \prod_{s=t+1}^{T} c_s = \prod_{s=1}^{T} c_s = C_T \tag{6.102}$$

*independent* of $t$. Hence the terms $C_t D_{t+1}$ cancel out of both the numerator and denominator of Eq. (6.101) and the exact reestimation equation is therefore realized.

It should be obvious that the above scaling procedure applies equally well to reestimation of the $\pi$ or $B$ coefficients. It should also be obvious that the scaling procedure of Eq. (6.95) need not be applied at every time instant $t$, but can be performed whenever desired, or whenever necessary (e.g., to prevent underflow). If scaling is not performed at some instant $t$, the scaling coefficients $c_t$ are set to 1 at that time, and all the conditions discussed above are then met.

The only real change to the HMM procedure because of scaling is the procedure for computing $P(\mathbf{O}|\lambda)$. We cannot merely sum up the $\hat{\alpha}_T(i)$ terms, because these are scaled already. However, we can use the property that

$$\prod_{t=1}^{T} c_t \sum_{i=1}^{N} \alpha_T(i) = C_T \sum_{i=1}^{N} \alpha_T(i) = 1. \tag{6.103}$$

Thus we have

$$\prod_{t=1}^{T} c_t \cdot P(\mathbf{O}|\lambda) = 1 \tag{6.104}$$

or

$$P(\mathbf{O}|\lambda) = \frac{1}{\prod_{t=1}^{T} c_t} \tag{6.105}$$

or

$$\log [P(\mathbf{O}|\lambda)] = \sum_{t=1}^{T} \log\, c_t. \tag{6.106}$$

Thus the log of $P$ can be computed, but not $P$, since it would be out of the dynamic range of the machine anyway.

Finally we note that when using the Viterbi algorithm to give the maximum likelihood state sequence, no scaling is required if we use logarithms as discussed in the alternate Viterbi implementation.



### 6.12.2 Multiple Observation Sequences

In Section 6.5 we discussed a form of HMM called the left-right or Bakis model, in which the state proceeds from state 1 at $t = 1$ to state $N$ at $t = T$ in a sequential manner (recall the model of Figure 6.8(b)). We have already discussed how a left-right model imposes constraints on the state-transition matrix, and the initial state probabilities Eqs. (6.45)–(6.48). However, the major problem with left-right models is that one cannot use a single observation sequence to train the model (i.e., for reestimation of model parameters). This is because the transient nature of the states within the model allows only a small number of observations for any state (until a transition is made to a successor state). Hence, to have sufficient data to make reliable estimates of all model parameters, one has to use multiple observation sequences ([18]).

The modification of the reestimation procedure is straightforward and is as follows. We denote the set of $K$ observation sequences as

$$\mathbf{O} = [\mathbf{O}^{(1)}, \mathbf{O}^{(2)}, \ldots, \mathbf{O}^{(K)}] \tag{6.107}$$

where $\mathbf{O}^{(k)} = (\mathbf{o}_1^{(k)} \mathbf{o}_2^{(k)} \ldots \mathbf{o}_{T_k}^{(k)})$ is the $k$th observation sequence. We assume each observation sequence is independent of every other observation sequence, and our goal is to adjust the parameters of the model $\lambda$ to maximize

$$P(\mathbf{O}|\lambda) = \prod_{k=1}^{K} P(\mathbf{O}^{(k)}|\lambda) \tag{6.108}$$

$$= \prod_{k=1}^{K} P_k. \tag{6.109}$$

Since the reestimation formulas are based on frequencies of occurrence of various events, the reestimation formulas for multiple observation sequences are modified by adding together the individual frequencies of occurrence for each sequence. Thus the modified reestimation formulas for $\bar{a}_{ij}$ and $\overline{b_j(\ell)}$ are

$$\bar{a}_{ij} = \frac{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k-1} \alpha_t^k(i) a_{ij} b_j(\mathbf{o}_{t+1}^{(k)}) \beta_{t+1}^k(j)}{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k-1} \alpha_t^k(i) \beta_t^k(i)} \tag{6.110}$$

and

$$\overline{b_j(\ell)} = \frac{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{\substack{t=1 \\ \text{s.t. } \mathbf{o}_t = v_\ell}}^{T_k-1} \alpha_t^k(i) \beta_t^k(i)}{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k-1} \alpha_t^k(i) \beta_t^k(i)} \tag{6.111}$$

and $\pi_i$ is not reestimated since $\pi_1 = 1$, $\pi_i = 0$, $i \neq 1$.

The proper scaling of Eqs. (6.110)–(6.111) is now straightforward since each observation sequence has its own scaling factor. The key idea is to remove the scaling factor from each term before summing. This can be accomplished by writing the reestimation equations in terms of the scaled variables, i.e.,

$$\bar{a}_{ij} = \frac{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k - 1} \hat{\alpha}_t^k(i) a_{ij} b_j(\mathbf{o}_{t+1}^k) \hat{\beta}_{t+1}^k(j)}{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k - 1} \hat{\alpha}_t^k(i) \hat{\beta}_t^k(i)}. \tag{6.112}$$

In this manner, for each sequence $\mathbf{O}^{(k)}$, the same scale factors will appear in each term of the sum over $t$ as appears in the $P_k$ term, and hence will cancel exactly. Thus using the scaled values of the $\alpha$s and $\beta$s results in an unscaled $\bar{a}_{ij}$. A similar result is obtained for the $\bar{b}_j(\ell)$ term.

### 6.12.3 Initial Estimates of HMM Parameters

In theory, the reestimation equations should give values of the HMM parameters that correspond to a local maximum of the likelihood function. A key question is, therefore, How do we choose initial estimates of the HMM parameters so that the local maximum is equal to or as close as possible to the global maximum of the likelihood function?

Basically there is no simple or straightforward answer. Instead, experience has shown that either random (subject to the stochastic and the nonzero value constraints) or uniform initial estimates of the $\pi$ and $A$ parameters are adequate for giving useful reestimates of these parameters in almost all cases. However, for the $B$ parameters, experience has shown that good initial estimates are helpful in the discrete symbol case and are essential (when dealing with multiple mixtures) in the continuous-distribution case. Such initial estimates can be obtained in a number of ways; these include (a) manual segmentation of the observation sequence(s) into states and averaging of observations within states, (b) maximum likelihood segmentation of observations and averaging, and (c) segmental $k$-means segmentation with clustering, etc. We discuss such segmentation techniques later in this chapter.

### 6.12.4 Effects of Insufficient Training Data

Another problem associated with training HMM parameters via reestimation methods is that the observation sequence used for training is, of necessity, finite ([30]). Thus there is always an inadequate number of occurrences of low-probability events (e.g., symbol occurrences within states) to give good estimates of the model parameters. By way of example, consider the case of a discrete observation HMM. Recall that the reestimation transformation of $\bar{b}_j(k)$, Eq. (6.40c), requires a count of the expected number of times in state $j$ and observing symbol $\mathbf{v}_k$ simultaneously. If the training sequence is so small that it does not have any occurrences of this event (i.e., $q_t = j$ and $\mathbf{o}_t = \mathbf{v}_k$), $b_j(k) = 0$ and



will stay 0 after reestimation. The resultant model would produce a zero probability result for any observation sequence that actually includes ($\mathbf{o}_t = \mathbf{v}_k$ and $q_t = j$). Such a singular outcome is obviously a consequence of the unreliable estimate that $b_j(k) = 0$ due to the insufficiency of the training set.

One solution to this problem is to increase the size of the training observation set. Often this is impractical. A second possible solution is to reduce the size of the model (e.g., number of states, number of symbols per state). Although this is always possible, often there are physical reasons why a given model is used, and therefore the model size cannot be changed. A third possible solution is to seek unconventional statistical estimation algorithms that can somehow enhance the reliability of the parameter estimates even based on limited training data. Deleted interpolation and parameter thresholding are two such alternatives. Since deleted interpolation is considered more an enhanced parameter estimation method, we discuss that subject in the next section.

The simplest way to handle the effects of insufficient training data is to add extra threshold constraints to the model parameters to ensure that no model parameter estimate falls below a specified level [18]. Thus, for example, we might specify the numeric floor, for a discrete symbol model, that

$$b_j(k) = \begin{cases} b_j(k), & \text{if } b_j(k) \geq \delta_b \\ \delta_b, & \text{otherwise} \end{cases} \tag{6.113a}$$

or, for a continuous distribution model, that

$$U_{jk}(r, r) = \begin{cases} U_{jk}(r, r), & \text{if } U_{jk}(r, r) \geq \delta_u \\ \delta_u, & \text{otherwise} \end{cases} \tag{6.113b}$$

When the numeric floor is invoked in the reestimation equations, all remaining parameters need to be rescaled so that the densities obey the required stochastic constraints. Such postprocessor techniques are thus considered implementational measures to combat the insufficient data problem and have been applied with good success to several problems in speech processing. The method of parameter thresholding has a justification from a Bayes statistics point of view. It can be shown that Eq. (6.113b) is, in fact, a maximum a posteriori (MAP) estimate of the variance under the assumption that the parameter prior $P(U_{jk}(r, r))$ is an informative one with uniform distribution and $(U_{jk}(r, r))_{\min} = \delta_u$ [31]. (See Section 6.13.)

### 6.12.5 Choice of Model

The remaining issues in implementing HMMs are the choice of type of model (ergodic or left-right or some other form), choice of model size (number of states), and choice of observation symbols (discrete or continuous, single or multimixture, choice of observation parameters). Unfortunately, there is no simple, theoretically correct way of making such choices. These choices must be made depending on the signal being modeled. With these comments, we end our discussion of the theoretical aspects of hidden Markov models and proceed to a discussion of how such models have been applied to the isolated word-recognition problem.

## 6.13 IMPROVING THE EFFECTIVENESS OF MODEL ESTIMATES

We discuss three methods that have been shown to be able to enhance the effectiveness of HMM model estimates for speech recognition. These are (1) deleted interpolation, (2) Bayesian adaptation, and (3) corrective training. The first two methods are motivated by the problem of insufficient data, while the last method has the unique objective of trying to reduce recognition errors directly.

### 6.13.1 Deleted Interpolation

When training data are insufficient, reliable and robust determination of HMM parameters cannot be accomplished. The HMM obtained by the Baum-Welch reestimation method, based on the maximum likelihood criterion, may be adequate in characterizing the training data, but for new data the match may be quite poor. One parameter estimation method that aims to improve the model reliability is the method of "deleted interpolation."

The concept involves combining two (or more) separately trained models, one of which is more reliably trained than the other. A scenario in which this can happen is the case when we use tied states which forces "different" states to share an identical statistical characterization, effectively reducing the number of parameters in the model. A model with tied states is often more robust than a model without tied states when trained on the same amount of data. But a model with tied states is also less precise than a model without tied states if the training data are sufficient. Therefore, the idea of combining the two models is to allow us to fall back to the more reliable model when the supposedly more precise model is, in fact, unreliable. A similar scenario occurs when context-independent (more robust) and context-dependent (more precise) phone models are used in large vocabulary recognition (see Chapter 8).

Let the two models be defined by the parameter sets $\lambda = (A, B, \pi)$ and $\lambda' = (A', B', \pi')$, respectively. The interpolated model, $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$, is obtained as

$$\bar{\lambda} = \epsilon\lambda + (1 - \epsilon)\lambda' \tag{6.114}$$

where $\epsilon$ represents the weighting of the parameters of the "full" model (with more detailed characterization of the observations) and $(1 - \epsilon)$ represents the weighting of the parameters of the reduced, but more reliable, model. A key issue is the determination of the optimal value of $\epsilon$, which is a strong function of the amount of training data. This is easy to see because as the amount of training data gets large, $\lambda$ becomes more reliable and we expect $\epsilon$ to tend to 1.0. Similarly, for small amounts of training data, $\lambda$ is unreliable and we expect $\epsilon$ to tend to 0.0 so as to fall back to the more reliable model $\lambda'$.

The solution to the determination of an optimal value for $\epsilon$ was provided by Jelinek and Mercer [30], who showed how the optimal $\epsilon$ could be estimated using the forward-backward algorithm by interpreting Eq. (6.114) as an expanded HMM of the type shown in Figure 6.12. Figure 6.12a shows the part of the state-transition structure related to the state $S$. Using Figure 6.12b, we can interpret the interpolated model of Eq. (6.114) as an



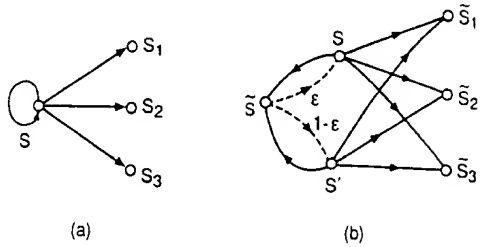



**Figure 6.12** Example of how the process of deleted interpolation can be represented using a state diagram.

expanded HMM in which each state is replaced by three states. The null transitions from the expanded state $\tilde{S}$ to $S$ and $S'$ have transition probabilities $\epsilon$ and $1 - \epsilon$, respectively. The transitions out of $S$ are characterized by those of $\lambda$ while those out of $S'$ are associated with those of $\lambda'$.

The expanded HMM interpretation suggests that the parameter $\epsilon$ can be optimally determined by the usual forward-backward algorithm. However, since the interpolation is designed to better predict unseen (future) data, rather than to account for the training data, determination of $\epsilon$ must be based on data that was not used in obtaining either of the two models, $\lambda$ and $\lambda'$. A key idea of deleted interpolation is thus to partition the training data $\mathcal{T}$ into two disjoint sets; that is, $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$. For example, one might consider a partition of the training set such that $\mathcal{T}_1$ is 90 percent of $\mathcal{T}$ and $\mathcal{T}_2$ is the remaining 10 percent of $\mathcal{T}$. Training set $\mathcal{T}_1$ is first used to train $\lambda$ and $\lambda'$. Training set $\mathcal{T}_2$ is then used to give an estimate of $\epsilon$, assuming $\lambda$ and $\lambda'$ are fixed. There are obviously a large number of ways in which such a partitioning can be accomplished, but one particularly simple one is to cycle $\mathcal{T}_2$ through the data. That is, the first partition uses the last 10 percent of the data as $\mathcal{T}_2$, the second partition uses the next-to-last 10 percent of the data as $\mathcal{T}_2$, etc. An interpretation of deleted interpolation is straightforward. If the unseen data fits the more elaborate model $\lambda$ well (thus validating the reliability of $\lambda$), the forward-backward algorithm would give a value of $\epsilon$ which is close to 1. Otherwise, the forward-backward algorithm gives a small value of $\epsilon$, indicating that the more reliable model $\lambda'$ is a better characterization of the new data than $\lambda$.

The technique of deleted interpolation has been successfully applied to a number of problems in speech recognition, including the estimation of trigram word probabilities for language models [13], and the estimation of HMM output probabilities for trigram phone models (to be discussed in Chapter 8 of this book).

### 6.13.2 Bayesian Adaptation

Another insufficient data situation occurs when we attempt to estimate a speaker-dependent model based on a limited amount of speaker-specific training data. An approach to this

problem is through speaker adaptation, in which a speaker-independent model, obtained by reliable training, is adapted to the particular talker using speaker-specific training data [31].

Speaker adaptation can be accomplished based on a Bayesian framework. Consider the HMM probability measure $P(\mathbf{O}|\lambda)$. If the HMM parameter $\lambda$ is assumed to be fixed but unknown, the maximum likelihood (ML) estimate for $\lambda$, given the training sequence $\mathbf{O}$, is obtained by solving the likelihood equation, i.e.

$$\frac{\partial}{\partial\lambda}P(\mathbf{O}|\lambda) = 0. \tag{6.115}$$

(Normally, the Baum-Welch reestimation algorithm is used to obtain certain stationary point solutions instead of directly solving for Eq. (6.115).) If $\lambda$ is assumed random with a priori distribution $P_0(\lambda)$, then the maximum a posteriori (MAP) estimate for $\lambda$ is obtained by solving

$$\frac{\partial}{\partial\lambda}P(\lambda|\mathbf{O}) = 0 \tag{6.116}$$

for the given training sequence $\mathbf{O}$. Using the Bayes theorem, we rewrite $P(\lambda|\mathbf{O})$ as

$$P(\lambda|\mathbf{O}) = \frac{P(\mathbf{O}|\lambda)P_0(\lambda)}{P(\mathbf{O})}. \tag{6.117}$$

The influence of the parameter prior $P_0(\lambda)$ in the solution process thus becomes explicit. Note that if the distribution is correctly chosen, the MAP solution attains minimum Bayes risk.

The parameter prior distribution characterizes the statistics of the parameters of interest before any measurement is made. If the prior distribution indicates no preference as to what the parameter values are likely to be, then the prior is called a noninformative prior (which is essentially constant for the entire parameter space). In this case, the MAP estimate obtained by solving Eq. (6.116) is identical to the ML estimate of Eq. (6.115). If we do have prior knowledge about the parameter values of the model, the incorporation of such prior knowledge in the form of a prior distribution would become important in the MAP estimate for minimum Bayes risk. This type of prior is often called an informative prior. Intuitively, if we know what the parameter values are likely to be before observations are made, we may be able to make good use of the data, which may be limited, to obtain a good model estimate. If this is true, the questions remaining are how to derive the informative prior and how to use it in obtaining the MAP estimate.

For mathematical tractability, conjugate priors are often used in Bayesian adaptation. A conjugate prior for a random vector is defined as the prior distribution for the parameters of the probability density function of the random vector, such that the posterior distribution $P(\lambda|\mathbf{O})$ and the prior distribution $P(\lambda)$ belong to the same distribution family for any sample observations $\mathbf{O}$. For example, it is well known that the conjugate prior for the mean of a Gaussian density is also a Gaussian density. In the following, we therefore discuss only the use of conjugate priors. We also discuss only the case of Bayesian adaptation of the Gaussian mean as it is sufficient to demonstrate the idea of Bayesian adaptation in dealing with small training set problems.



Let us focus on a Gaussian mixture component $\mathcal{N}(\mu, \sigma^2)$ in a mixture density HMM. We use one-dimensional observations for simplicity. Assume the mean $\mu$ is random with prior distribution $P_0(\mu)$ and the variance $\sigma^2$ is known and fixed. It can be shown that the conjugate prior for $\mu$ is also Gaussian; that is, if we assume $P_0(\mu)$ to be the conjugate prior of $\mu$, then $P_0(\mu)$ is Gaussian. Thus, let us denote the mean and variance of the prior for $\mu$ by $\rho$ and $\tau^2$, respectively. The MAP estimate for the mean parameter $\mu$ in Bayesian adaptation, from a set of $n$ training observations, is given by

$$\hat{\mu}_{\text{MAP}} = \frac{n\tau^2}{\sigma^2 + n\tau^2}\bar{\mathbf{o}} + \frac{\sigma^2}{\sigma^2 + n\tau^2}\rho \tag{6.118}$$

where $\bar{\mathbf{o}}$ is the sample mean of the $n$ training data. The interpretation of Eq. (6.118) is as follows. If there are no training data presented, $n = 0$ and the best estimate of $\mu$ is simply the mean $\rho$ of the prior distribution of the $\mu$ parameter. When training data are collected and used, the MAP estimate becomes a weighted average of the prior mean $\rho$ and the sample mean of the presented observations, $\bar{\mathbf{o}}$. Ultimately, $n \to \infty$ and the best estimate of $\mu$ is, as expected, the sample mean $\bar{\mathbf{o}}$. It should also be noted that if the prior variance $\tau^2$ is much larger than $\sigma^2/n$, the MAP estimate in Eq. (6.118) is essentially the ML estimate, $\bar{\mathbf{o}}$, which corresponds to the case of using noninformative priors.

A key question is, How do we determine $\rho$ and $\tau^2$? In practice, these prior parameters have to be estimated from a collection of speaker-dependent (or multispeaker) models, or from a speaker-independent model with mixture distributions in each state. For example, $\rho$ and $\tau^2$ can be estimated by

$$\rho = \sum_{m=1}^{M} c_m \rho_m \tag{6.119a}$$

and

$$\tau^2 = \sum_{m=1}^{M} c_m (\rho_m - \rho)^2 \tag{6.119b}$$

where $c_m$ is the weight assigned to the $m^{\text{th}}$ model (or the $m^{\text{th}}$ mixture component in the corresponding state of a mixture density speaker-independent HMM) and $\rho_m$ is the mean of the corresponding $m^{\text{th}}$ model (or mixture component). When using a speaker-independent Gaussian mixture HMM, the weight $c_m$ is basically the mixture gain for the $m^{\text{th}}$ mixture component, and the estimates of Eq. (6.119) are the ML estimates of the mean and variance parameters of $\mu$ before any speaker-specific training data are observed.

The concept of Bayesian adaptation based on conjugate priors can be applied to other parameters as well. The adaptation method can be shown to provide good parameter estimates even when the number of speaker-specific training tokens is extremely limited. Experiments have shown that large improvements in recognition accuracy are obtained with the Bayesian adaptation method, compared to direct training, particularly when only a small number of training tokens are available [30].

### 6.13.3 Corrective Training

In statistical pattern recognition, the minimum Bayes' risk is the theoretical recognizer performance bound, conditioned on the exact knowledge of both the class prior $P(v)$ and the conditional distributions $P(\mathbf{O}|v)$. When both distributions are not known exactly, and the classifier needs to be designed based on a finite training set, there are several ways to try to reduce the error rate. One method is based on the theoretical link between discriminant analysis and distribution estimation [32]. The idea is to design a classifier (discriminant function) such that the minimum classification error rate is attained on the training set. In particular, we wish to design a classifier that uses estimates of $P(v)$ and $P(\mathbf{O}|v)$, and that achieves a minimum error rate for the training set in the same way a discriminant function is designed. The reason for using the HMM, $P(\mathbf{O}|\lambda_v)$, for modeling $P(\mathbf{O}|v)$, as opposed to other discriminant functions, is to exploit the strengths of the HMMs—consistency, flexibility and computational ease.

Bahl et al. [33] were the first to propose an error-correction strategy, which they named corrective training, to specifically deal with the misclassification problem. Their training algorithm was motivated by analogy with an error-correction training procedure for linear classifiers. In their proposed method, the observation distribution is of a discrete type, $B = [b_i(k)]$, where $b_i(k)$ is the probability of observing a vector quantization code index (acoustic label) $k$ when the HMM source is in state $i$. Each $b_i(k)$ is obtained via the forward-backward algorithm as the weighted frequency of occurrence of the code index. The corrective training algorithm of Bahl et al. works as follows. First, use a labeled training set to estimate the parameters of the HMMs $\Lambda = \{\lambda_v\}$ with the forward-backward algorithm. For each utterance $\mathbf{O}$, labeled as $v$, for example, evaluate $P(\mathbf{O}|\lambda_v)$ for the correct class $v$ and $P(\mathbf{O}|\lambda_w)$ for each incorrect class $w$. (The evaluation of likelihood for the incorrect classes need not be exhaustive.) For every utterance where $\log P(\mathbf{O}|\lambda_w) > \log P(\mathbf{O}|\lambda_v) - \Delta$, where $\Delta$ is a prescribed threshold, modify $\lambda_v$ and $\lambda_w$ according to the following mechanism: (1) Apply the forward-backward algorithm to obtain estimates $b_i'(k)$ and $b_i''(k)$, using the labeled utterance $\mathbf{O}$ only, for the correct class $v$ and incorrect class $w$, respectively; (2) Modify the original $b_i(k)$ in $\lambda_v$ to $b_i(k) + \gamma b_i'(k)$ and the $b_i(k)$ in $\lambda_w$ to $b_i(k) - \gamma b_i''(k)$. When the state labels are tied for certain models, the above procedure is equivalent to replacing the original $b_i(k)$ by $b_i(k) + \gamma(b_i'(k) - b_i''(k))$. The prescribed adaptation parameter, $\gamma$, controls the "rate of convergence" and the threshold, $\Delta$, defines the "near-miss" cases. This corrective training algorithm therefore focuses on those parts of the model that are most important for word discrimination, a clear difference from the maximum likelihood principle.

Bahl et al. reported that the corrective training procedure worked better (in isolated word-recognition tasks) than models obtained using the maximum mutual information or the conditional maximum likelihood criterion. The method, however, is primarily experimental.

Several other forms of discriminative training were also proposed by Katagiri et al. [34] together with a framework for the analysis of related training/learning ideas for minimizing recognition errors. The discriminative training method described in Sec. 5.6.3 can be applied to HMM training without difficulty. The corrective training algorithm of Bahl et al. can be shown to be just one possible choice for the minimization of a prescribed risk function.



## 6.14 MODEL CLUSTERING AND SPLITTING

One of the basic assumptions in statistical modeling is that the variability in the observations from an information source can be modeled by the assumed statistical distributions. For speech recognition, the source could be a single word, a subword unit like a phoneme, or a word sequence. Because of variability in the production (e.g., accents, speed of talking), or the processing (e.g., transmission distortion, noise), it is often expedient to consider using more than a single HMM to characterize the source. There are two motivations behind this multiple HMM approach. First, lumping together all the variability from inhomogeneous data sources leads to unnecessarily complex models, often yielding lower modeling accuracy. Second, some of the variability, or rather the inhomogeneity in the source data, may be known *a priori*, thus warranting separate modeling of the source data sets.

Several generalized clustering algorithms exist, such as the $k$-means clustering algorithm, the generalized Lloyd algorithm as is widely used in vector quantizer designs [35], and the greedy growing algorithm found in set partition or decision tree designs [36], all of which are suitable for the purpose of separating inconsistent training data so that each divided subgroup becomes more homogeneous and therefore is better modeled by a single HMM. The nearest-neighbor rule required in these clustering algorithms is simply to assign an observation sequence $\mathbf{O}$ to cluster $i$ if

$$P(\mathbf{O}|\lambda_i) = \max_j P(\mathbf{O}|\lambda_j) \tag{6.120}$$

where $\lambda_j$s denote the models of the clusters. Successful application of the model clustering algorithms to the speech-recognition problem, using the straightforward maximum likelihood criterion, has been reported.

An alternative to model clustering is to arbitrarily subdivide a given speech source into a large number of subclasses with specialized characteristics and then consider a generalized procedure for model merging based on source likelihood considerations. By way of example, for large vocabulary speech recognition we often try to build specialized units (context sensitive) for recognition. For example, we could consider building units that are a function of the sound immediately preceding the unit (left-context) and the sound immediately following the unit (right-context). There are on the order of 10,000 such units in English. Many of the units are functionally almost identical. The problem is how to determine which pairs of units should be merged (so that the number of model units is made more manageable and the variance of the parameter estimate is reduced). To get ideas, consider two distinct models, $\lambda_a$ and $\lambda_b$, corresponding to training observation sets $\mathbf{O}_a$ and $\mathbf{O}_b$, and the merged model $\lambda_{a+b}$, corresponding to the merged observation sets $\{\mathbf{O}_a, \mathbf{O}_b\}$. We can then compute the change in entropy (i.e., loss of information) resulting from the

merged model as

$$\begin{aligned}\Delta H_{ab} &= H_a + H_b - H_{a+b} \\ &= -P(\mathbf{O}_a|\lambda_a)\log P(\mathbf{O}_a|\lambda_a) - P(\mathbf{O}_b|\lambda_b)\log P(\mathbf{O}_b|\lambda_b) \\ &\quad + P(\{\mathbf{O}_a, \mathbf{O}_b\}|\lambda_{a+b})\log P(\{\mathbf{O}_a, \mathbf{O}_b\}|\lambda_{a+b}).\end{aligned} \tag{6.121}$$

Whenever $\Delta H_{ab}$ is small enough, it means that the change in entropy resulting from merging the models will not affect system performance (at least on the training set) and the models can be merged. The question of how small is acceptable is dependent on specific applications. This model merging technique has been used successfully by Lee [37] to create a generalized set of triphone models for large vocabulary speech recognition.

## 6.15 HMM SYSTEM FOR ISOLATED WORD RECOGNITION

To illustrate the techniques discussed in this chapter, consider using HMMs to build an isolated word recognizer ([38]). Assume we have a vocabulary of $V$ words to be recognized and that each word is to be modeled by a distinct HMM. Further assume, for simplicity of notation, that for each word in the vocabulary we have a training set of $K$ utterances of the word (spoken by one or more talkers) where each utterance constitutes an observation sequence, of some appropriate representation of the (spectral and/or temporal) characteristics of the word. To do isolated word speech recognition, we must perform the following:

1. For each word $v$ in the vocabulary, we must build an HMM $\lambda_v$—that is, we must estimate the model parameters $(A, B, \pi)$ that optimize the likelihood of the training set observation vectors for the $v$th word.
2. For each unknown word to be recognized, the processing shown in Figure 6.13 must be carried out, namely, measurement of the observation sequence $\mathbf{O} = \{\mathbf{o}_1\, \mathbf{o}_2 \ldots \mathbf{o}_T\}$, via a feature analysis of the speech corresponding to the word; followed by calculation of model likelihoods for all possible models, $P(\mathbf{O}|\lambda_v)$, $1 \le v \le V$; followed by selection of the word whose model likelihood is highest—that is,

$$v^* = \arg \max_{1 \le v \le V} [P(\mathbf{O}|\lambda_v)]. \tag{6.122}$$

The probability computation step is generally performed using the Viterbi algorithm (i.e., the maximum likelihood path is used) and requires on the order of $V \cdot N^2 \cdot T$ computations. For modest vocabulary sizes, e.g., $V = 100$ words, with an $N = 5$ state model, and $T = 40$ observations for the unknown word, a total of $10^5$ computations is required for recognition (where each computation is a multiply, and add, and a calculation of observation density, $b(\mathbf{o})$). Clearly this amount of computation is modest compared to the capabilities of most modern signal processor chips.



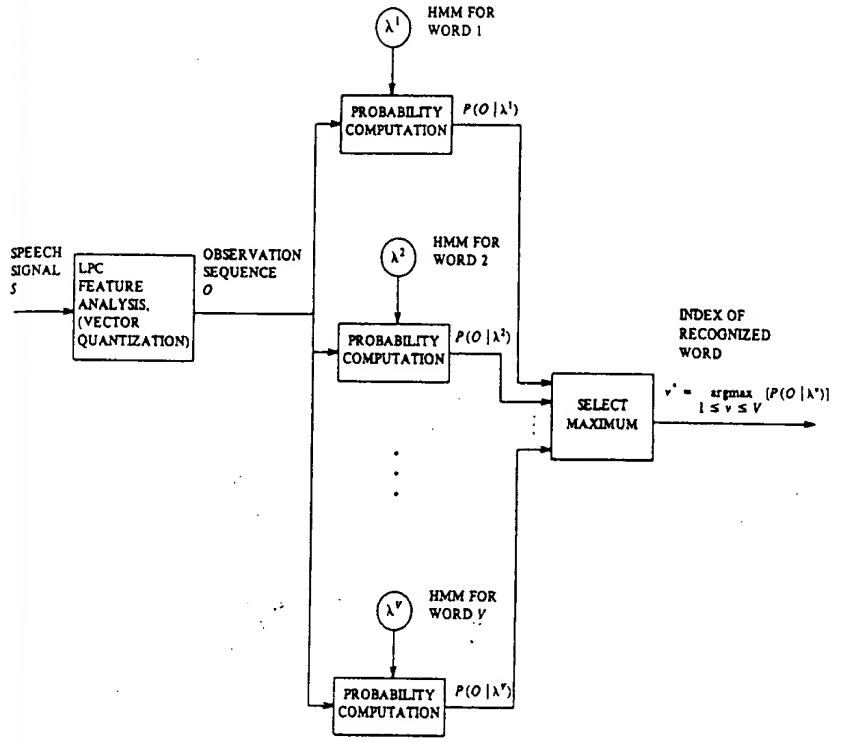



Figure 6.13 Block diagram of an isolated word HMM recognizer (after Rabiner [38]).

### 6.15.1 Choice of Model Parameters

We now return to the issue that we have raised several times in this chapter—namely, How do we select the type of model, and how do we choose the parameters of the selected model? For isolated word recognition with a distinct HMM designed for each word in the vocabulary, it should be clear that a left-right model is more appropriate than an ergodic model, since we can then associate time with model states in a fairly straightforward manner. Furthermore, we can envision the physical meaning of the model states as distinct sounds (e.g., phonemes, syllables) of the word being modeled.

The issue of the number of states to use in each word model leads to two schools of thought. One idea is to let the number of states correspond roughly to the number of sounds (phonemes) within the word—hence, models with from 2 to 10 states would be appropriate [18]. The other idea is to let the number of states correspond roughly to the average number of observations in a spoken version of the word, the so-called Bakis model

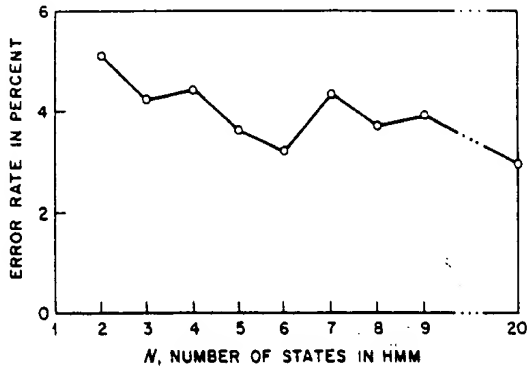



**Figure 6.14** Average word error rate (for a digits vocabulary) versus the number of states $N$ in the HMM (after Rabiner et al. [18]).

[11]. In this manner each state corresponds to an observation interval—i.e., about 10–15 ms for the standard methods of analysis. In the results to be described later in this section, we use the former approach. Furthermore, we restrict each word model to have the same number of states; this implies that the models will work best when they represent words with the same number of sounds.

To illustrate the effect of varying the number of states in a word model, Figure 6.14 shows a plot of average word error rate versus $N$, for the case of recognition of isolated digits (i.e., a 10-word vocabulary). It can be seen that the error is somewhat insensitive to $N$, achieving a local minimum at $N = 6$; however, differences in error rate for values of $N$ close to 6 are small.

The next issue is the choice of observation vector and the way it is represented. As discussed in Chapter 3, possibilities include LPC-derived weighted cepstral coefficients and weighted cepstral derivatives or (for autoregressive HMMs) the autocorrelation coefficients as the observation vectors for continuous models; for discrete symbol models we use a codebook to generate the discrete symbols. For the continuous models we use as many as $M = 64 \sim 256$ mixtures per state; for the discrete symbol models we use codebooks with as many as $M = 512 \sim 1024$ code words. Also, for the continuous models, it has been found that it is more convenient and sometimes preferable to use diagonal covariance matrices with several mixtures, rather than fewer mixtures with full covariance matrices. The reason for this is simple—namely, the difficulty in performing reliable reestimation of the off diagonal components of the covariance matrix from the necessarily limited training data. Figure 6.15 illustrates the need for using mixture densities for modeling observation vectors (eighth-order cepstral vectors derived from LPC with log energy appended as the ninth vector component). Figure 6.15 shows a comparison of marginal distributions $b_j(\mathbf{o})$ against a histogram of the actual observations within a state (as determined by a maximum likelihood segmentation of all the training observations into states). The observation vectors



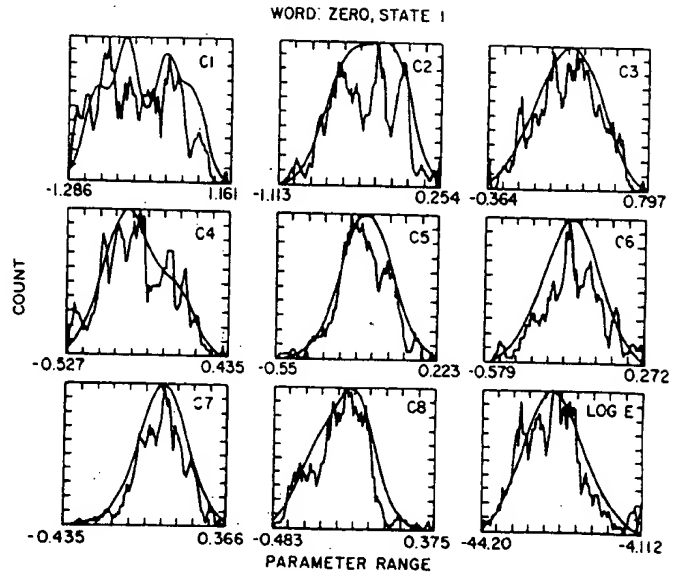



**Figure 6.15** Comparison of estimated density (jagged contour) and model density (smooth contour) for each of the nine components of the observation vector (eight cepstral components, one log energy component) for state 1 of the digit zero (after Rabiner et al. [38]).

are ninth order, and the model density uses $M = 5$ mixtures. The covariance matrices are constrained to be diagonal for each individual mixture. The results of Figure 6.15 are for the first model state of the word "zero." The need for values of $M > 1$ is clearly seen in the histogram of the first parameter (the first cepstral component) which is inherently multimodal; similarly, the second, fourth, and eighth cepstral parameters show the need for more than a single Gaussian component to provide good fits to the empirical data. Many of the other parameters appear to be well fitted by a single Gaussian; in some cases, however, even $M = 5$ mixtures do not provide a sufficiently good fit.

Another experimentally verified fact about the HMM is that it is important to limit some of the parameter estimates to prevent them from becoming too small. For example, for the discrete symbol models, the constraint that $b_j(k)$ be greater than or equal to some minimum value $\epsilon$ is necessary to ensure that even when the $k$th symbol never occurred in some state $j$ in the training observation set, there is always a finite probability of its occurrence when scoring an unknown observation set. To illustrate this point, Figure 6.16 shows a curve of average word error rate versus the parameter $\epsilon$ (on a log scale) for a standard word-recognition experiment. It can be seen that over a very broad range ($10^{-10} \leq \epsilon \leq 10^{-3}$) the average error rate remains at about a constant value; however,

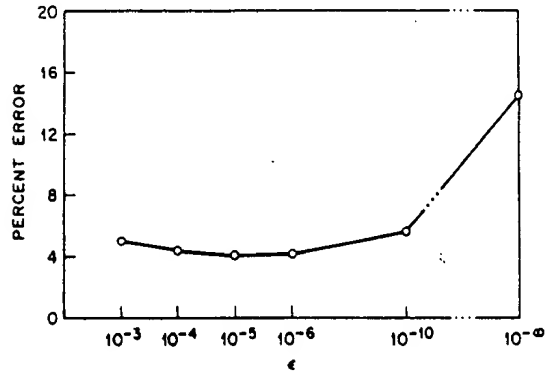



**Figure 6.16** Average word error rate as a function of the minimum discrete density value $\epsilon$ (after Rabiner et al. [18]).

when $\epsilon$ is set to 0 (i.e., $10^{-\infty}$), then the error rate increases sharply. Similarly, for continuous densities it is important to constrain the mixture gains $c_{jm}$ as well as the diagonal covariance coefficients $U_{jm}(r, r)$ to be greater than or equal to some minimum values (we use $10^{-4}$ in all cases) [18].

### 6.15.2 Segmental K-Means Segmentation into States

In this chapter, we have emphasized that good initial estimates of the parameters of the $b_j(\mathbf{o}_t)$ densities are essential for rapid and proper convergence of the reestimation formulas. Hence a procedure for providing good initial estimates of these parameters was devised and is shown in Figure 6.17. The training procedure is a variant on the well-known $K$-means iterative procedure for clustering data.

We assume that we have a training set of observations (the same as is required for parameter reestimation), and an initial estimate of all model parameters. However, unlike the one required for reestimation, the initial model estimate can be chosen randomly or on the basis of any available model appropriate to the data.

Following model initialization, the set of training observation sequences is segmented into states, based on the current model $\lambda$. This segmentation is achieved by finding the optimum state sequence, via the Viterbi algorithm, and then backtracking along the optimal path. This procedure is illustrated in Figure 6.18, which shows a log-energy plot, an accumulated log-likelihood plot, and a state segmentation for one occurrence of the word "six." The states correspond roughly to the sounds in the spoken word "six."

The result of segmenting each of the training sequences is, for each of the $N$ states, a maximum likelihood estimate of the set of the observations that occur within each state $j$ according to the current model. In the case where we are using discrete symbol densities, each of the observation vectors within a state is coded using the $M$-code-word codebook,



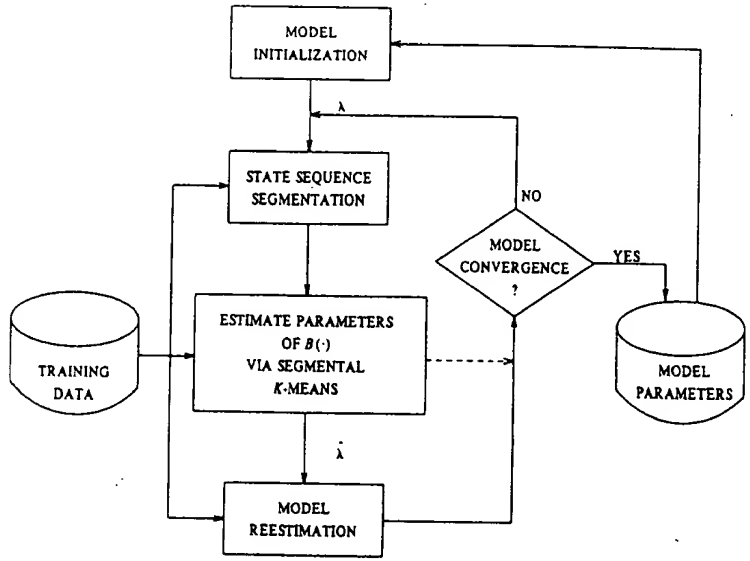



**Figure 6.17** The segmental $k$-means training procedure used to estimate parameter values for the optimal continuous mixture density fit to a finite number of observation sequences (after Rabiner et al. [38]).

and the updated estimate of the $b_j(k)$ parameters is

$$\hat{b}_j(k) = \text{number of vectors with codebook index } k \text{ in state } j \text{ divided by the number of vectors in state } j.$$

When we are using continuous observation densities, a segmental $K$-means procedure is used to cluster the observation vectors within each state $j$ into a set of $M$ clusters (using a Euclidean distortion measure), where each cluster represents one of the $M$ mixtures of the $b_j(\mathbf{o}_t)$ density. From the clustering, an updated set of model parameters is derived as follows:

$\hat{c}_{jm}$ = number of vectors classified in cluster $m$ of state $j$ divided by the number of vectors in state $j$

$\hat{\boldsymbol{\mu}}_{jm}$ = sample mean of the vectors classified in cluster $m$ of state $j$

$\hat{\mathbf{U}}_{jm}$ = sample covariance matrix of the vectors classified in cluster $m$ of state $j$.

Based on this state segmentation, updated estimates of the $a_{ij}$ coefficients can be obtained by counting the number of transitions from state $i$ to $j$ and dividing it by the number of transitions from state $i$ to any state (including itself).

An updated model $\hat{\lambda}$ is obtained from the new model parameters, and the formal reestimation procedure is used to reestimate all model parameters. The resulting model is then compared to the previous model (by computing a distance score that reflects the

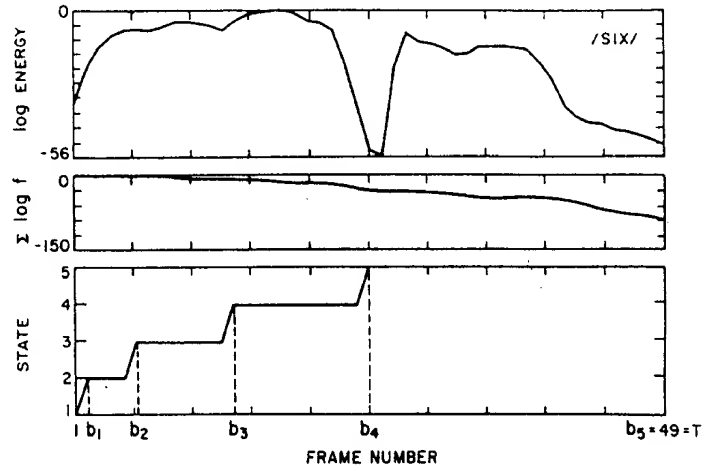



**Figure 6.18** Plots of (a) log energy; (b) accumulated log likelihood; and (c) state assignment for one occurrence of the word "six" (after Rabiner et al. [38]).

statistical similarity of the HMMs). If the model distance score exceeds a threshold, then the old model $\lambda$ is replaced by the new (reestimated) model $\bar{\lambda}$, and the overall training loop is repeated. If the model distance score falls below the threshold, then model convergence is assumed and the final model parameters are saved.

### 6.15.3 Incorporation of State Duration into the HMM

In Section 6.9 we discussed the theoretically correct method of incorporating state duration information into the mechanics of the HMM ([39]). We also showed that the cost of including duration density was rather high; namely a $D^2$-fold increase in computation and a $D$-fold increase in storage. Using a value of $D = 25$ (as is required for word recognition), the cost of the increased computation tended to make the techniques not worth using. Thus the following alternative procedure was formulated for incorporating state duration information into the HMM.

For this alternative procedure, the state duration probability $p_j(d)$ was measured directly from the segmented training sequences used in the segmental $K$-means procedure of the previous section. Hence the estimates of $p_j(d)$ are strictly heuristic ones. A typical set of histograms of $p_j(d)$ for a five-state model of the word "six" is shown in Figure 6.19. (In this figure the histograms are plotted versus normalized duration $(d/T)$, rather than absolute duration $d$.) The first two states account for the initial /s/ in "six"; the third state accounts for the transition to the vowel /i/; the fourth state accounts for the vowel; and the fifth state accounts for the stop and the final /s/ sound.

The way in which the heuristic duration densities were used in the recognizer was



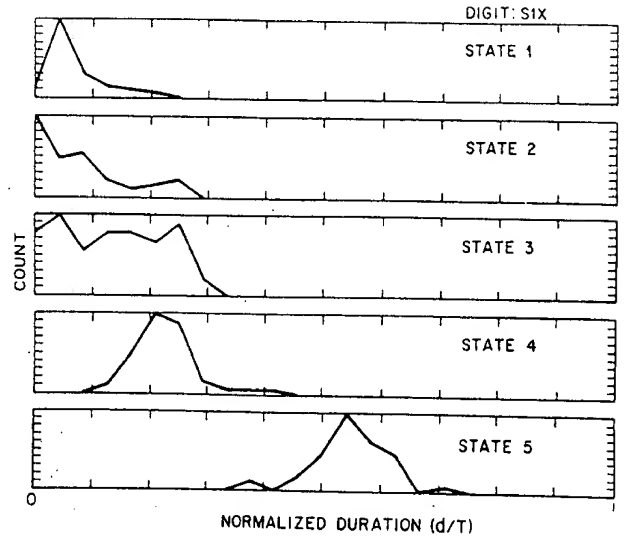



**Figure 6.19** Histograms of the normalized duration density for the five states of the digit "six" (after Rabiner et al. [38]).

as follows. First the normal Viterbi algorithm is used to give the best segmentation of the observation sequence of the unknown word into states via a backtracking procedure. The duration of each state is then measured from the state segmentation. A postprocessor then increments the log-likelihood score of the Viterbi algorithm, by the quantity

$$\log \hat{P}(\mathbf{q}, \mathbf{O}|\lambda) = \log P(\mathbf{q}, \mathbf{O}|\lambda) + \alpha_d \sum_{j=1}^{N} \log [p_j(d_j)] \tag{6.123}$$

where $\alpha_d$ is a scaling multiplier on the state duration scores, and $d_j$ is the duration of state $j$ along the optimal path as determined by the Viterbi algorithm. The incremental cost of the postprocessor for duration is essentially negligible, and experience has shown that recognition performance is essentially as good as that obtained using the theoretically correct duration model.

### 6.15.4 HMM Isolated-Digit Performance

We conclude this section on isolated word recognition using HMMs by giving a set of performance results (in terms of average word error rate) on the task of recognizing isolated digits in a speaker-independent manner. For this task, a training set consisting of 100 occurrences of each digit by 100 talkers (i.e., a single occurrence of each digit per talker) was used. Half the talkers were male and half were female. For testing the

TABLE 6.1. Average Digit Error Rates for Several Recognizers and Evaluation Sets

| Recognizer Type | Evaluation Set: Original Training | TS2 | TS3 | TS4 |
|---|---|---|---|---|
| LPC/DTW | 0.1 | 0.2 | 2.0 | 1.1 |
| LPC/DTW/VQ | – | 3.5 | – | – |
| HMM/VQ | – | 3.7 | – | – |
| HMM/CD | 0 | 0.2 | 1.3 | 1.8 |
| HMM/AR | 0.3 | 1.8 | 3.4 | 4.1 |

algorithm, we used the initial training set, as well as three other independent test sets with the following characteristics:

TS2 The same 100 talkers as were used in the training; 100 occurrences of each digit

TS3 A new set of 100 talkers (50 male, 50 female); 100 occurrences of each digit

TS4 Another new set of 100 talkers (50 male, 50 female); 100 occurrences of each digit

The results of the recognition tests are given in Table 6.1. The recognizers are the following:

LPC/DTW Conventional template-based recognizer using dynamic time warping (DTW) alignment

LPC/DTW/VQ Conventional recognizer with vector quantization of the feature vectors ($M = 64$)

HMM/VQ HMM recognizer with $M = 64$ codebook

HMM/CD HMM recognizer using continuous density model with $M = 5$ mixtures per state

HMM/AR HMM recognizer using autoregressive observation density

It can be seen that, when using a VQ, the performance of the isolated word recognizer degrades in both the conventional and HMM modes. It can also be seen that the performances of the conventional template-based recognizer, and the HMM recognizer with a continuous density model are comparable. Finally Table 6.1 shows that the autoregressive density HMM gives poorer performance than the standard mixture density model.

## 6.16 SUMMARY

In this chapter we have attempted to present the theory of hidden Markov models from the simplest concepts (discrete Markov chains) to the most sophisticated models (variable duration, continuous density models). It has been our purpose to focus on physical explanations of the basic mathematics; hence we have avoided long, drawn-out proofs or derivations of the key results, and concentrated primarily on trying to interpret the meaning of the math, and how it could be implemented in practice in real-world systems. We have also attempted to illustrate one application of the theory of HMMs to a simple problem in speech recognition—namely, isolated word recognition.



## REFERENCES

[1] L.E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *Ann. Math. Stat.*, 37: 1554-1563, 1966.

[2] L.E. Baum and J.A. Egon, "An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology," *Bull. Amer. Meteorol. Soc.*, 73: 360-363, 1967.

[3] L.E. Baum and G.R. Sell, "Growth functions for transformations on manifolds," *Pac. J. Math.*, 27 (2): 211-227, 1968.

[4] L.E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Ann. Math. Stat.*, 41 (1): 164-171, 1970.

[5] L.E. Baum, "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes," *Inequalities*, 3: 1-8, 1972.

[6] J.K. Baker, "The dragon system—An overview," *IEEE Trans. Acoustics, Speech, Signal Proc.*, ASSP-23 (1): 24-29, February 1975.

[7] F. Jelinek, "A fast sequential decoding algorithm using a stack," *IBM J. Res. Develop.*, 13: 675-685, 1969.

[8] L.R. Bahl and F. Jelinek, "Decoding for channels with insertions, deletions, and substitutions with applications to speech recognition," *IEEE Trans. Information Theory*, IT-21: 404-411, 1975.

[9] F. Jelinek, L.R. Bahl, and R.L. Mercer, "Design of a linguistic statistical decoder for the recognition of continuous speech," *IEEE Trans. Information Theory*, IT-21: 250-256, 1975.

[10] F. Jelinek, "Continuous speech recognition by statistical methods," *Proc. IEEE*, 64: 532-536, April 1976.

[11] R. Bakis, "Continuous speech word recognition via centisecond acoustic states," in *Proc. ASA Meeting* (Washington, DC), April 1976.

[12] F. Jelinek, L.R. Bahl, and R.L. Mercer, "Continuous speech recognition: Statistical methods," in *Handbook of Statistics, II*, P.R. Krishnaiad, Ed. Amsterdam, The Netherlands: North-Holland, 1982.

[13] L.R. Bahl, F. Jelinek, and R.L. Mercer, "A maximum likelihood approach to continuous speech recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-5: 179-190, 1983.

[14] J.D. Ferguson, "Hidden Markov Analysis: An Introduction," in *Hidden Markov Models for Speech*, Institute for Defense Analyses, Princeton, NJ, 1980.

[15] A.J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE Trans. Information Theory*, IT-13: 260-269, April 1967.

[16] G.D. Forney, "The Viterbi algorithm," *Proc. IEEE*, 61: 268-278, March 1973.

[17] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc.*, 39 (1): 1-38, 1977.

[18] S.E. Levinson, L.R. Rabiner, and M.M. Sondhi, "An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition," *Bell System Tech. J.*, 62 (4): 1035–1074, April 1983.

[19] L.A. Liporace, "Maximum likelihood estimation for multivariate observations of Markov sources," *IEEE Trans. Information Theory*, IT-28 (5): 729-734, 1982.

[20] B.H. Juang, "Maximum likelihood estimation for mixture multivariate stochastic observations of Markov chains," *AT&T Tech. J.*, 64 (6): 1235-1249, July-Aug. 1985.

[21] B.H. Juang, S.E. Levinson, and M.M. Sondhi, "Maximum likelihood estimation for multivariate mixture observations of Markov chains," *IEEE Trans. Information Theory*, IT-32 (2): 307-309, March 1986.

[22] A.B. Poritz, "Linear predictive hidden Markov models and the speech signal," in *Proc. ICASSP 82* (Paris, France), 1291-1294, May 1982.

[23] B.H. Juang and L.R. Rabiner, "Mixture autoregressive hidden Markov models for speech signals," *IEEE Trans. Acoustics, Speech, Signal Proc.*, ASSP-33 (6): 1404-1413, December 1985.

[24] M J. Russell and R.K. Moore, "Explicit modeling of state occupancy in hidden Markov models for automatic speech recognition," in *Proc. ICASSP 85* (Tampa, FL), 5-8, March 1985.

[25] S.E. Levinson, "Continuously variable duration hidden Markov models for automatic speech recognition," *Computer Speech and Language*, 1 (1): 29-45, March 1986.

[26] L.R. Bahl, P.F. Brown, P.V. deSouza, and L.R. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Proc. ICASSP 86* (Tokyo, Japan), 49-52, April 1986.

[27] Y. Ephraim, A. Dembo, and L.R. Rabiner, "A minimum discrimination information approach for hidden Markov modeling," *IEEE Trans. Information Theory*, 35 (5): 1001-1003, September 1989.

[28] Y. Ephraim and L. Rabiner, "On the Relations Between Modeling Approaches for Speech Recognition," *IEEE Trans. Information Theory*, 36 (2): 372-380, March 1990.

[29] B.H. Juang and L.R. Rabiner, "A probabilistic distance measure for hidden Markov models," *AT&T Tech. J.*, 64 (2): 391-408, February 1985.

[30] F. Jelinek and R.L. Mercer, "Interpolated estimation of Markov source parameters from sparse data," in *Pattern Recognition in Practice*, E.S. Gelesma and L.N. Kanal, Eds. Amsterdam, The Netherlands: North-Holland, 1980, 381-397.

[31] C.-H. Lee, C.-H. Lin, and B.H. Juang, "A study on speaker adaptation of the parameters of continuous density hidden Markov models," *IEEE Trans. Signal Processing*, 39 (4): 806–841, April 1991.

[32] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York, 1973.

[33] L.R. Bahl, P.F. Brown, P.V. deSouza, and R.L. Mercer, "A new algorithm for the estimation of hidden Markov model parameters," *Proc. ICASSP 88*, 493–496, New York, April 1988.

[34] S. Katagiri, C.H. Lee, and B.H. Juang, "New Discriminative Training Algorithms Based on the Generalized Probabilistic Descent Method," *Proc. 1991 Workshop Neural Networks for Signal Processing*, 299–308, IEEE Signal Processing Society, Princeton, September 1991.



[35] Y. Linde, A. Buzo, and R.M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Communications*, COM-28: 84–95, January 1980.

[36] J. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, *Classification and Regression Trees*, Wadsworth & Brooks, Monterey, CA, 1984.

[37] K.F. Lee, *Automatic Speech Recognition—The Development of the SPHINX System*, Kluwer Academic Publishers, Boston, 1989.

[38] L.R. Rabiner, B.H. Juang, S.E. Levinson, and M.M. Sondhi, "Recognition of Isolated Digits Using Hidden Markov Models With Continuous Mixture Densities," *AT&T Tech. J.*, 64 (6): 1211–1234, July-Aug. 1985.

[39] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE*, 77 (2): 257–286, February 1989.